



Proyecto Fin de Carrera
Ingeniería Informática
Curso 2011/2012

Planificación de la navegación de robots en entornos dinámicos

María Teresa Lorente Cebrián

Director: Luis Montano Gella

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior
Universidad de Zaragoza

Diciembre 2011

Planificación de la navegación de robots en entornos dinámicos

RESUMEN

La navegación autónoma de robots en entornos dinámicos sigue siendo un desafío en el campo de la Robótica. Muchas aplicaciones que utilizan robots móviles requieren autonomía en su movilidad para poder realizar otras tareas como, por ejemplo, el proporcionar un servicio o el llevar a cabo una exploración en un terreno desconocido. En la mayoría de escenarios de aplicación pueden aparecer personas u objetos móviles que dificulten la navegación y entorpezcan el desarrollo de su actividad.

Desde el punto de vista del robot, la tarea más importante es garantizar su integridad evitando la colisión con otros obstáculos a su alrededor mientras navega hacia su objetivo en el menor tiempo posible, siempre y cuando las restricciones del entorno y las del propio robot lo permitan. En cuanto a los entornos dinámicos, estos plantean otros problemas como la localización, la detección y seguimiento de otros objetos móviles, la predicción de su estado futuro y la planificación de los movimientos del robot en un entorno cambiante.

La información que se dispone del entorno y que supone la base para el sistema de planificación desarrollado viene facilitada por un sensor láser a bordo del robot. Esta información junto con el valor de odometría que se obtiene del propio robot serán necesarias para la construcción del mapa del entorno y la resolución del problema de la localización. Para la identificación de los objetos móviles y su seguimiento se emplea el *Filtro de Kalman Extendido (FKE)*.

La novedad que aporta este trabajo es una técnica de planificación *robocéntrica* para la navegación de robots que calcula comandos de movimiento directamente en el espacio de control, el *espacio de velocidad*, sin restringir los comandos disponibles. La técnica se basa en un modelo en el que los obstáculos estáticos y móviles se representan en un mismo marco de trabajo. Las restricciones cinemáticas y dinámicas del vehículo se tratan explícitamente en el modelo, al que denominamos *Espacio de Velocidad Dinámico (EVD)*, calculando comandos de movimiento aplicables a vehículos reales sin aproximaciones.

El razonar sobre este modelo lleva al planificador a identificar una *situación* de entre un conjunto finito de situaciones definidas en un *Árbol de Situaciones*. Se establece una *jerarquía* de condiciones relacionadas con estas situaciones para obtener la estrategia de navegación. La *seguridad* del vehículo y la evitación de otros obstáculos se consideran como el objetivo principal, dejando como secundario el obtener el movimiento más rápido hacia la posición destino, aunque también es relevante para establecer la estrategia de navegación.

Finalmente, la técnica se evalúa en simulación para analizar el rendimiento del sistema en diferentes escenarios. Además, se realizan experimentos con robots para mostrar las características y limitaciones que aparecen en situaciones reales.

Índice general

| | |
|---|-----------|
| 1. Introducción | 1 |
| 1.1. Objetivos | 1 |
| 1.2. Trabajo relacionado | 2 |
| 1.3. Estructura de la memoria | 4 |
| 2. Modelado del entorno en el <i>EVD</i> | 5 |
| 2.1. Modelado de los objetos móviles en el <i>EVD</i> | 6 |
| 2.2. Fusión de los objetos móviles en el <i>EVD</i> | 7 |
| 2.3. Objetos estáticos en el <i>EVD</i> | 8 |
| 3. Estrategias para la toma de decisión | 11 |
| 4. Estrategias de navegación | 15 |
| 4.1. Planificación de la navegación | 16 |
| 4.1.1. MovimientoLibre | 16 |
| 4.1.2. Pasar Por Delante | 17 |
| 4.1.3. Pasar Por Detrás | 18 |
| 4.1.4. Dentro de la Banda de Colision | 20 |
| 4.1.5. Varios obstáculos | 20 |
| 5. Evaluación de la planificación de la navegación | 23 |
| 5.1. Caos | 23 |
| 5.2. Carriles | 24 |
| 5.3. Adelantamiento | 26 |
| 6. Experimentación real | 31 |
| 6.1. Preparación de los experimentos | 31 |
| 6.2. Experimento 1: navegación por un hall | 32 |
| 6.3. Experimento 2: navegación por un pasillo | 34 |
| 7. Conclusiones y trabajo futuro | 37 |
| 7.1. Conclusiones | 37 |
| 7.2. Limitaciones y problemas encontrados | 38 |
| 7.3. Trabajo futuro | 39 |

| | |
|--|-----------|
| A. Herramientas utilizadas | 41 |
| B. Algunos conceptos | 43 |
| B.1. Robots no holonómicos | 43 |
| B.2. Transformaciones | 43 |
| C. Tratamiento de la información del sensor | 45 |
| C.1. Filtro de Kalman Extendido | 45 |
| C.2. Segmentación de la información estática | 47 |
| D. Modelado del entorno | 49 |
| D.1. Fusión de objetos móviles | 49 |
| D.2. Objetos estáticos | 50 |
| E. Estrategias de navegación | 53 |
| E.1. Variables de decisión | 53 |
| E.2. Estrategias de navegación | 54 |
| E.2.1. MovimientoLibre | 54 |
| E.2.2. Pasar Por Delante | 56 |
| E.2.3. Pasar Por Detrás | 58 |
| E.2.4. Dentro de la Banda de Colisión | 60 |
| E.2.5. Varios obstáculos | 63 |
| F. Simulaciones | 69 |
| F.1. Caos | 69 |
| F.2. Carriles | 71 |
| F.3. Adelantamiento | 72 |
| G. Experimentos | 77 |
| G.1. Experimento 1 | 77 |
| G.2. Experimento 2 | 77 |
| H. Distribución temporal | 85 |
| Índice de figuras | 89 |
| Bibliografía | 91 |

Capítulo 1

Introducción

En este primer capítulo de la memoria se presentan los objetivos que marcan la finalidad de este Proyecto Fin de Carrera (PFC), así como el trabajo relacionado con diferentes aspectos del mismo y la organización de la memoria.

1.1. Objetivos

El objetivo general de este proyecto es el desarrollo de un sistema de planificación automática de movimientos para robots móviles para entornos dinámicos, es decir, aquellos en los que de forma natural aparecen personas y otros vehículos en movimiento, y cuya información dinámica puede utilizarse para planificar mejor el propio movimiento del robot.

Se trata del desarrollo del software que permita la integración sobre la arquitectura informática embarcada en los robots móviles del grupo de Robótica de la Universidad de Zaragoza disponibles en el Departamento. Se utilizará la plataforma de desarrollo [Player/Stage] para programar y poner a punto los programas en simulación, pero utilizando la dinámica de los robots reales. Este mismo software se probará en los robots reales ajustando los parámetros, para validar el sistema realizado. El lenguaje utilizado es C/C++ sobre Player/Stage.

A continuación se señalan las tareas realizadas en el proyecto y que, por tanto, marcaron los hitos del mismo:

- Estudio de los conceptos detrás del *EVD* y refinamiento del modelo. Esta fase supuso la lectura del trabajo donde se define el modelo, en [Owen y Montano (2005)]. Se parte de un modelo ya implementado en Matlab que debe traducirse a C/C++. Durante este proceso se estudia en detalle el modelo y se detectan casos no tratados en la implementación, lo que lleva a rediseñar el software para obtener un modelo del entorno más completo. Se define también una técnica de fusión de objetos móviles sobre el *EVD*.
- Análisis del mapeo de un obstáculo en el *EVD*, cuya representación se denomina *Objeto Dinámico de Velocidad (ODV)*. Se debe estudiar el comportamiento dinámico de un obstáculo sobre el *EVD*, es decir, cómo evoluciona la representación del objeto móvil en

el *EVD* conforme se mueve en el *Espacio de Trabajo (ET)*. Esto permitirá identificar las situaciones en las que se encontrará el robot, teniendo en cuenta la posición y velocidad del obstáculo. Este mismo proceso se lleva posteriormente a escenarios con más de un obstáculo.

- Implementación de los algoritmos generales de planificación en entornos dinámicos. A partir de las situaciones identificadas en la etapa anterior se determina un conjunto de estrategias que resuelvan dichas situaciones.
- Incorporación al modelo de la información estática. Inicialmente, el tratamiento de la información estática no formaba parte del alcance de este proyecto. Sin embargo, fue necesario tenerla en cuenta dado que los experimentos se realizan en entornos de interior. Para su definición sobre el *EVD* se emplea un concepto diferente al de objetos dinámicos, lo que implica la actualización de las estrategias ya definidas para considerar esta nueva información.
- Análisis del software sobre construcción del mapa de estáticos y dinámicos, localización y *FKE*. En esta fase se trató de entender de forma general las técnicas existentes para tratar la información obtenida por el láser, prestando especial atención a la parte dedicada al *FKE*.
- Puesta a punto en simulación de los algoritmos mediante Player/Stage.
- Puesta a punto sobre los robots Pioneer reales, ajustando los parámetros del sistema. Se evalúa el sistema en un entorno real con otros vehículos y personas en movimiento.

El resultado final esperado es una aplicación en la que se ponga de manifiesto la potencia del sistema implementado y pueda evaluarse su robustez, limitaciones y aplicabilidad en un entorno real.

Cabe resaltar el hecho de que los resultados obtenidos con el trabajo realizado se han presentado en un artículo enviado a una revista de primer nivel en Robótica, *Autonomous Robots*. El artículo está en fase de revisión, [Owen, Lorente y Montano].

1.2. Trabajo relacionado

Tradicionalmente, los diferentes enfoques relacionados con la navegación autónoma se han clasificado en métodos de planificación deliberativa y reactiva. Los primeros son costosos ya que su propósito es calcular inicialmente una trayectoria completa hasta el objetivo. [Latombe] y [Choset y otros] proporcionan descripciones detalladas de los métodos deliberativos. Los métodos reactivos tienen una visibilidad hacia delante reducida debido a que únicamente calculan el siguiente comando de movimiento del robot como por ejemplo, [Fox y otros] y [Ho y Simmons]. Estos métodos hacen uso de la reactividad del robot para tratar el dinamismo del entorno pero no suelen tener en cuenta las trayectorias de los objetos para anticipar el espacio libre. Como consecuencia, las trayectorias resultantes pueden cambiar rápidamente

y el robot puede verse en una situación de atrapamiento.

Para tratar este problema, las aproximaciones reactivas se combinan con técnicas de planificación global, como por ejemplo, [Stachniss y Burgard], [Minguez y Montano], [Fraichard] y [Brock y Kathib]. En lugar de calcular sólo el siguiente comando de movimiento, estas técnicas calculan varios pasos dependiendo del tiempo horizonte disponible.

Uno de los problemas que surgen al enfrentarse a ambientes dinámicos es la capacidad para describir de forma eficiente el dinamismo del entorno en un espacio de dominio reducido para disminuir el tiempo de respuesta mientras se toman decisiones de movimiento en tiempo real. [Fiorini y Shiller] tienen en cuenta explícitamente las velocidades de los obstáculos. La idea es representar los obstáculos directamente en el espacio de velocidad del robot para calcular el conjunto de velocidades que llevan a colisión, aunque no consideran las restricciones de movimiento del robot.

El aspecto de la seguridad en la planificación de movimientos en entornos dinámicos se considera en [Kuffner y LaValle]. El problema requiere razonar sobre la importancia del comportamiento de los objetos móviles en el futuro así como la cantidad de tiempo que este futuro se tiene en cuenta (horizonte de tiempo).

En [Owen y Montano (2005)] se propuso un modelo que consideraba el espacio de velocidad continuo como el espacio de control, en el que todos los comandos de velocidad conducentes a trayectorias en espacio libre podían ser escogidas. alguna de estas velocidades son finalmente seleccionadas en cada periodo de tiempo, empleando un criterio que conduce al robot de forma rápida y segura a los objetivos, proporcionando maniobras suaves, siempre que es posible. En este trabajo se introdujo la idea de calcular movimientos para que el robot pasara antes del objeto móvil o para que frenara dejando pasar al objeto. Además, las restricciones cinemáticas y dinámicas se reflejan explícitamente en el modelo.

El trabajo realizado en este PFC extiende dicho modelo y contribuye con estrategias de navegación para robots en entornos dinámicos como un método heurístico para calcular comandos de velocidad utilizando ese modelo. Las decisiones se toman directamente sobre el espacio de velocidad del robot, que es el espacio de control natural para elegir los comandos para navegar. Además, las restricciones cinemáticas y dinámicas del robot están implícitas en la estrategia seleccionada. El sistema calcula movimientos dentro del periodo de control, pero emplea un tiempo horizonte que se extiende al campo de visión del sensor que percibe el entorno, en nuestro caso, un láser. Sin embargo, sólo se aplica el comando calculado para el siguiente periodo de tiempo, permitiendo que el sistema pueda adaptarse a cambios en el escenario en tiempo real.

1.3. Estructura de la memoria

El contenido de la memoria se divide en varios capítulos que engloban todo el trabajo realizado y una serie de anexos que amplían la documentación sobre algunos capítulos o proporcionan información técnica.

El capítulo 2, que sigue a éste de introducción, trata sobre el modelado del entorno estático y dinámico del robot en el espacio de velocidades. El tercer capítulo presenta el proceso de toma de decisión general y en el capítulo 4 se exponen las diferentes situaciones y estrategias obtenidas como resultado del trabajo realizado. En el siguiente capítulo se evalúa el sistema de planificación en simulación, mientras que los experimentos reales se muestran en el capítulo 6. Las conclusiones se recogen en el capítulo 7 junto con las dificultades encontradas durante la realización del PFC y se plantean mejoras y líneas de continuación de este proyecto.

En cuanto a los anexos, estos se han estructurado de la siguiente manera: El primero de los apéndices trata sobre la plataforma Hardware y Software empleada para el desarrollo del proyecto. El siguiente explica qué es un robot no holonómico y las transformaciones entre el sistema del mundo y el del robot. En el tercer anexo se describe el *FKE* y el proceso de segmentación de la información estática. El apéndice D se centra en aspectos relacionados con el modelado del entorno en el *EVD*. Los detalles sobre la planificación de la navegación se muestran en el anexo E. En los dos apéndices siguientes se presenta un análisis de las simulaciones y experimentos, respectivamente. El último apéndice resume la distribución temporal del proyecto.

Por último, se incluye un índice de figuras y la bibliografía consultada a lo largo del desarrollo del proyecto, tanto la que se hace referencia en este documento como la empleada para completar formación.

Capítulo 2

Modelado del entorno en el *EVD*

En este capítulo se explican las técnicas empleadas para modelar las diferentes partes del entorno del robot.

Las estrategias implementadas se basan en tomar decisiones en un espacio de velocidad robocéntrico. Para eso, los objetos estáticos y dinámicos tienen que modelarse en este espacio de manera que la información representada sea adecuada para elaborar las estrategias de planificación.

En [Owen y Montano (2005)] se presenta la técnica empleada para reflejar el dinamismo del entorno. El modelo presentado se basa en representar el entorno dinámico del robot en el espacio de velocidades dinámico, al que llamamos *EVD*. Se construye utilizando el concepto de *tiempo estimado de llegada*, que define el *tiempo a colisión* y el *tiempo para escapar de una colisión*, representados implícitamente sobre el modelo y que se utilizan, entre otras piezas de información, para calcular las maniobras más rápidas y seguras para el robot, permitiendo la planificación de movimientos con un horizonte de visibilidad mayor que el correspondiente a un periodo de muestreo. Poner este concepto en práctica lleva a un modelo en el que los comandos de movimiento del robot pueden calcularse directamente sobre el *EVD*, permitiendo que sean aplicados inmediatamente al robot. Razonar en términos de velocidad proporciona una forma más fácil y natural de tomar decisiones durante la navegación del robot en un entorno dinámico.

Dicho modelo se aplica a robots no holonómicos (apéndice B), sujetos a restricciones cinemáticas y dinámicas, por lo que se imponen algunas restricciones para simplificar la complejidad del problema, pero que no suponen una limitación para el método propuesto:

1. El robot se mueve describiendo trayectorias rectas y circulares. Se trata de una restricción común impuesta a robots no holonómicos asumiendo que el comando de velocidad aplicado es constante durante un periodo de tiempo y que el proceso se repite en el siguiente periodo. De esta manera, las restricciones cinemáticas y dinámicas se tienen en cuenta, y se pueden conseguir trayectorias complejas durante la navegación, como clotoides y anti-clotoides, al conectar pequeños tramos de trayectorias circulares.

2. Los obstáculos se mueven siguiendo trayectorias rectas a velocidad constante. Dado que el modelo se recalcula en cada periodo de tiempo, el movimiento de los objetos podría considerarse como una secuencia de trayectorias cortas en línea recta.
3. En el modelo, los objetos móviles se representan como cuadrados. Esto facilita el cálculo del modelo y proporciona un margen de seguridad para calcular los tiempos de colisión. Además, para los objetos móviles no es necesario usar su forma exacta, basta con considerar la zona barrida por ellos cuando se mueven.

2.1. Modelado de los objetos móviles en el *EVD*

La representación de los objetos móviles alrededor del robot se lleva a cabo calculando aquellas trayectorias que el robot podría seguir de forma segura y aquellas que lo llevarían a colisionar con los objetos. Estas trayectorias se expresan como una secuencia de velocidades en el *EVD*. Los *tiempos a colisión* obtenidos son información deducida durante el proceso de mapeo desde el *Espacio de Configuración (EC)* al *EVD*, el cuál se construye como se indica a continuación.

El sistema de referencia local del robot se usa para representar la información requerida en el modelo (representación robocéntrica). La figura 2.1a ilustra el *Espacio de Trabajo (ET)* con un robot y un objeto moviéndose a velocidad constante y siguiendo una trayectoria recta. La figura 2.1b refleja esta situación en el *EC* en el instante k . Muestra la *banda de colisión (BC)* (área barrida por el objeto móvil a lo largo de una línea recta) y un objeto móvil en dos posiciones (x_o^1 y x_o^2), que se refieren a las posiciones que el robot alcanzaría cuando, siguiendo una trayectoria circular r_j , el objeto justo acabara de pasarlo en el instante t_{1j} (punto P_{c1j}), o bien, el robot cruzara antes de que el objeto lo alcance en el instante t_{2j} (punto P_{c2j}) escapando de una colisión. El cálculo de estos puntos y tiempos es la base del modelo. Las velocidades y tiempos de colisión asociados a esos puntos ($P_{c1j} \rightarrow (w_1, v_1, t_{1j})$ y $P_{c2j} \rightarrow (w_2, v_2, t_{2j})$) se calculan como en [Owen y Montano (2005)]. Estos cálculos se extienden a todo el espacio *Velocidad-Tiempo* considerando un rango de radios de curvatura lo suficientemente amplio para formar una superficie sobre este espacio que sea significativa del entorno. La figura 2.1c ilustra el *Espacio Velocidad-Tiempo del Objeto Dinámico (EVTOD)* resultante para la situación en la figura 2.1a. En él se representan todas aquellas velocidades que podrían llevar al robot a una colisión si se mantuvieran durante un tiempo. Para simplificar el cálculo de la trayectoria en tiempo real, la técnica que se desarrolla no trabaja directamente sobre el *EVTOD*, sino con su proyección sobre el plano (w, v) , el *EVD*.

La figura 2.1d muestra esta proyección. A la zona oscura se le llamará *Objeto Dinámico de Velocidad, ODV*. Su interpretación en cuanto a comandos permitidos es clara. Sus valores del límite inferior (contorno inferior) representan las velocidades máximas que el robot puede tomar para permitir que el objeto pase antes que él (es decir, V_1 para la trayectoria r_j) y los valores del límite superior (contorno superior) se refieren a las velocidades mínimas que el robot debería llevar para pasar antes de que el objeto pueda atropellarlo (es decir, V_2).

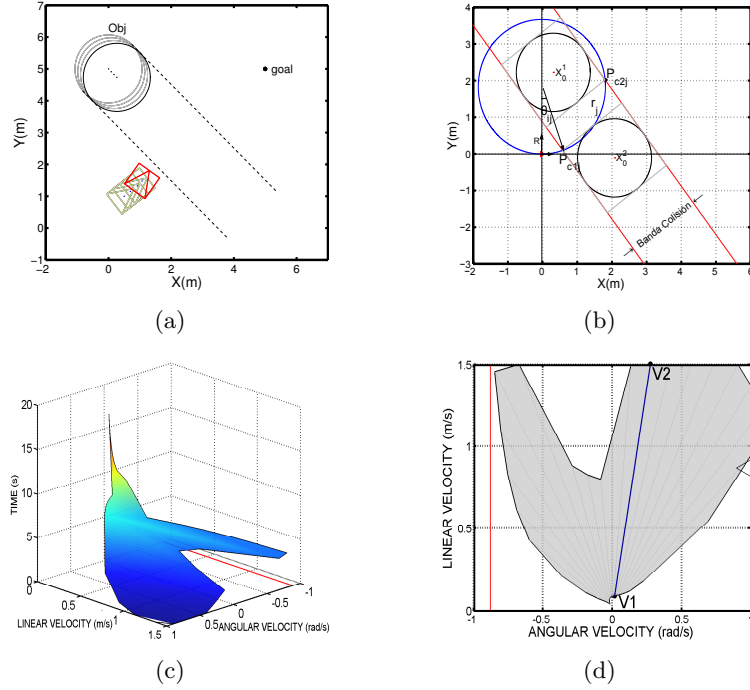


Figura 2.1: (a) Espacio de Trabajo, (b) banda de colisión, trayectoria r_j y puntos de colisión P_{c1j} and P_{c2j} en el Espacio de Configuración robocéntrico (R), (c) un Objeto Dinámico en el Espacio de Tiempo-Velocidad ($EVTOD$), (d) proyección EVD sobre el plano (w, v) .

Seleccionando velocidades fuera del ODV se asegura que el robot no colisionará durante el tiempo considerado por el cálculo. Además, el tamaño de los $ODVs$ en el EVD permite distinguir qué obstáculos suponen mayor peligro para el robot ya que reflejan tanto la cercanía o lejanía de los obstáculos como la velocidad que llevan con respecto al robot.

Es importante observar que los caminos circulares en el EC están representados como líneas rectas en el EVD y cuyo origen está en el origen del plano (w, v) . Esta característica facilitará la planificación de los movimientos para solucionar el problema de la navegación.

2.2. Fusión de los objetos móviles en el EVD

Siempre que haya obstáculos móviles situados en la zona de percepción del robot el sistema los fusiona para reducir la complejidad del problema de planificación de la navegación.

Esta fase del proyecto conllevó el análisis de diferentes técnicas que permitieran obtener un modelo representativo de todo el entorno y no supusieran una limitación en cuanto al conjunto de movimientos permitidos. Se hace una visión general en el apéndice D.1.

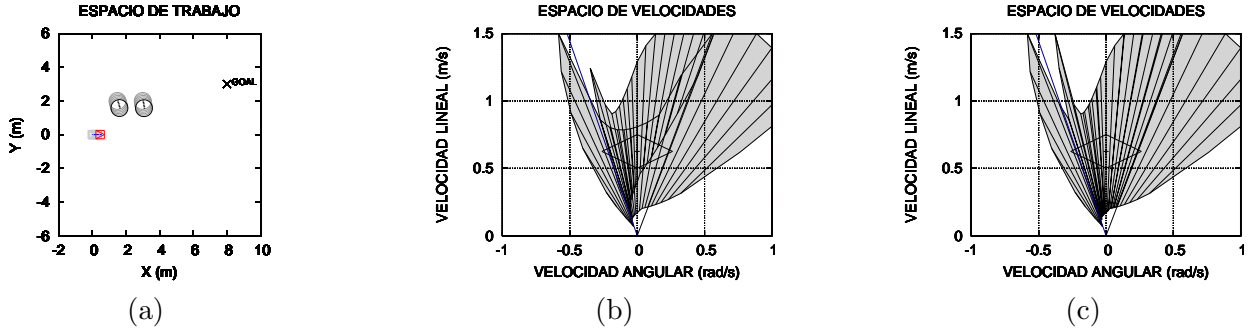


Figura 2.2: (a) Situación con dos objetos móviles en el *ET* (b) Misma situación en el *EVD* (c) *ODV* resultante de la fusión de los dos objetos móviles en el *EVD*

El proceso de fusión trata de determinar si las trayectorias que conducen al robot de forma segura con respecto a un obstáculo, lo hacen también para el resto de obstáculos. La figura 2.2 ilustra esta situación con dos obstáculos en el *ET* y el correspondiente *EVD* antes y después de la fusión. El algoritmo fusiona los objetos móviles calculando los límites del contorno a partir de las velocidades más extremas de todos los objetos involucrados. Así, se obtiene una representación completa del entorno que permite aplicar la misma técnica de navegación, sin importar el número de objetos visibles en el escenario.

2.3. Objetos estáticos en el *EVD*

Los objetos estáticos identificados en el entorno surgen de un proceso de segmentación de los puntos del láser clasificados como estáticos. Se explica en el apéndice C.2. Su representación sobre el *EVD* requiere una formulación diferente a la utilizada para tratar los dinámicos.

Dado que los objetos estáticos no tienen velocidad, su mapeo en el *EVD* utilizando directamente el método mostrado en la sección 2.1 nos llevaría a prohibir todas las velocidades de caminos circulares que pudieran llevar al robot a colisión, permitiendo sólo el comando de velocidad cero como único libre. El resultado es que grandes objetos estáticos como pasillos, paredes o límites de carreteras ocuparían gran parte de las velocidades en el *EVD*, restringiendo la navegación del robot.

Esta dificultad se puede resolver fácilmente teniendo en cuenta una mínima distancia de seguridad entre el robot y el objeto estático, $D_{seguridad}$, que se define como la distancia que el robot necesita para alcanzar la velocidad cero desde su velocidad actual. Si la longitud del camino circular desde la posición del robot al objeto estático es mayor que $D_{seguridad}$, entonces la trayectoria circular sobre el *EVD* se mapea como *libre*, es decir, todas las velocidades son libres de colisión. En otro caso, se calcula el límite de velocidad máximo que el robot podría llevar para poder parar antes de colisionar, $V_{frenado}$. El comando de velocidad encontrado se mapea en el *EVD* y cualquier velocidad mayor será también velocidad de colisión.

La figura 2.3 refleja el resultado de mapear un objeto estático de tipo pasillo en el *EVD*. En la figura 2.3a, se muestran tres trayectorias circulares que podrían corresponder a giros a derecha e izquierda hechos por el robot. En la figura 2.3b aparecen mapeadas en el *EVD*. Las velocidades de colisión cubren una zona mayor del lado izquierdo del *EVD* debido a la proximidad del robot con la pared en el lado derecho del pasillo. Esta técnica permite expresar en el mismo marco de trabajo tanto los objetos estáticos como los móviles.

En el anexo D.2 se amplía la información sobre el tratamiento que se hace de los estáticos y se muestran las ecuaciones implementadas para el modelado sobre el *EVD*.

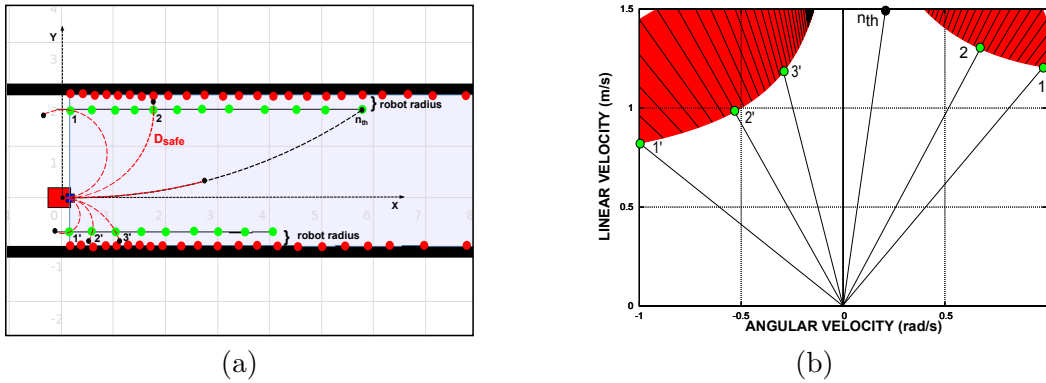


Figura 2.3: (a) Modelado de un pasillo en el *EVD*. Los puntos rojos corresponden a puntos del láser, los verdes son los puntos después de engordar las paredes con el radio del robot. (b) El pasillo mapeado en el *EVD*, mostrando la máxima velocidad alcanzable en cada camino.

Capítulo 3

Estrategias para la toma de decisión

En este capítulo se pretende introducir el proceso general de toma de decisión de un robot que tiene que evitar obstáculos móviles.

El problema es similar a una situación en la que un peatón quiere cruzar una calle en la que hay coches circulando. El peatón debe decidir cómo hacerlo de forma segura y en nuestro caso, el robot tiene que decidir lo mismo. En una situación de este tipo, la decisión principal que debe tomar el robot es si pasa antes que el obstáculo o bien espera hasta que el obstáculo haya pasado primero. Normalmente, las técnicas propuestas para navegar en entornos dinámicos reducen el número de maniobras posibles del robot o simplemente resuelven el problema de evitación de colisión haciendo que el robot disminuya la velocidad o pare permitiendo que el obstáculo pase primero. Sin embargo, estas soluciones restringen la capacidad de maniobra del robot e incrementan el tiempo empleado en alcanzar el objetivo.

Con el trabajo desarrollado en este proyecto se presenta una técnica de planificación de movimientos que intenta aprovechar toda la capacidad de maniobra del robot sin restringir las velocidades a un conjunto pequeño, mientras se intenta alcanzar el objetivo en el menor tiempo posible. El criterio principal es preservar la seguridad tanto de los objetos involucrados como la del robot. No obstante, otros criterios se consideran simultáneamente o jerárquicamente, como el movimiento más rápido hacia el objetivo, cambios suaves en los movimientos, etc.

La figura 3.1 muestra esta idea. Básicamente, hay dos estrategias principales: pasar por delante del objeto (*RobotAntes*) y pasar más tarde que el objeto (*RobotDespues*). Cada estrategia se ejecuta de forma diferente dependiendo de si el robot está antes (*AntesBanda*), dentro (*DentroBanda*) o después (*DespuesBanda*) de la *banda de colisión* del obstáculo, que es la zona barrida por él a lo largo de una línea recta. En cada una de estas etapas se ejecutan estrategias de movimiento diferentes, que dependen a su vez de otras variables tales como la localización y velocidad relativas entre el robot y los objetos, y el número de obstáculos presentes.

Se pueden aplicar varias técnicas para solucionar el problema. La empleada aquí se basa en el modelo de espacio de velocidad robocéntrico explicado en el capítulo 2, y en la detección y evaluación de la situación actual de entre un conjunto de situaciones definidas. Estas situaciones se representan de manera formal en las hojas del árbol de decisión de la figura 3.2. En el capítulo 4 se explica cada una de las situaciones y las estrategias de navegación asociadas.

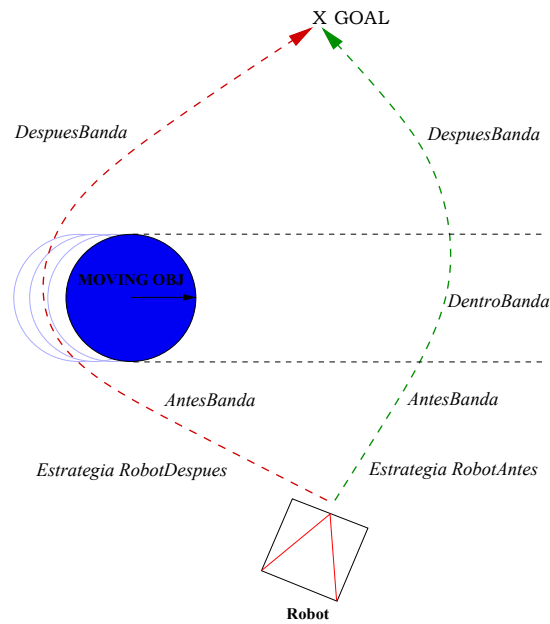


Figura 3.1: La trayectoria en verde corresponde a cuando el robot pasa antes que el obstáculo móvil y la trayectoria en rojo, cuando el robot pasa después.

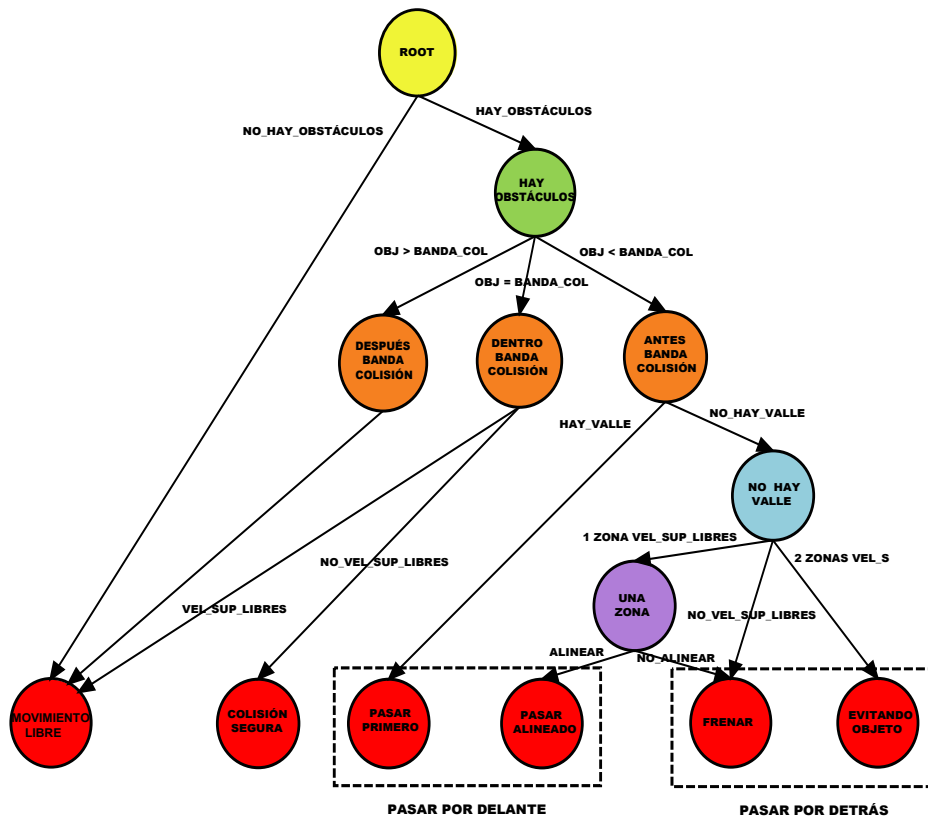


Figura 3.2: *Árbol de Situaciones*. Las hojas representan todas las situaciones que se pueden identificar. Cada una de ellas tiene una estrategia de navegación asociada.

Capítulo 4

Estrategias de navegación

A lo largo de este capítulo se explican las estrategias de navegación para las diferentes situaciones definidas en el *árbol de situaciones* de la figura 3.2.

Para poder llegar a definir el conjunto final de estrategias fue necesario identificar primero las posibles situaciones en las que podía verse involucrado el robot. Se trabajó inicialmente en C/C++ con un único obstáculo para sacar las primeras conclusiones sobre las diferentes situaciones que se iban presentando así como las distintas soluciones que se podían plantear para resolverlas. A continuación, se trabajó con escenarios con varios obstáculos para comprobar que las situaciones eran también adecuadas actualizando el conjunto de estrategias. Lo mismo se hizo al incorporar la información estática.

Varias técnicas propuestas en la literatura sobre navegación definen un índice de rendimiento a ser optimizado. Este índice se calcula como combinación lineal de diferentes criterios, a los que se les asocia un peso con el fin de obtener el comando óptimo de acuerdo a este índice. Este tipo de técnicas proporcionan comandos en tiempo real para mover los robots. Simplifican y resuelven el problema pero calculan comandos que son dependientes de los pesos, seleccionados de forma empírica. Dado que el número de parámetros y variables involucrados en la navegación es alta, especialmente en entornos dinámicos y densos, el hecho de dar peso a los criterios considerándolos al mismo nivel podría llevar a trayectorias no seguras o a movimientos lentos. Por eso, se sugiere el utilizar los criterios de manera jerárquica como una solución mejor para este tipo de escenarios. El método propuesto se basa en esta idea.

A partir del entorno percibido por el láser a bordo del robot y que se mapea en el espacio de velocidad (*EVD*, capítulo 2) se identifican diferentes situaciones. Para cada una de ellas se selecciona una estrategia de movimiento, que dependerá de varias variables deducidas del *EVD* y del *ET*. En el anexo E.1 se definen y analizan estas variables, denominadas *variables de decisión*. El planificador de la navegación, que calcula los comandos de movimiento en función de estas variables, se describe en el apartado 4.1.

4.1. Planificación de la navegación

La técnica de planificación de navegación descrita en este apartado se centra en calcular las trayectorias del robot para alcanzar un objetivo en un entorno dinámico, siguiendo los criterios de seguridad y rapidez de movimientos.

[Renaud y Fourquet] calcularon secuencias de movimientos de máxima aceleración como controles extremos para robots con control de aceleración para alcanzar trayectorias óptimas en tiempo en espacio libre. [Balkcom y Mason] demostraron que utilizando velocidades máximas lineales y angulares para vehículos diferenciales como controles extremos en espacio libre se producen trayectorias óptimas en tiempo para alinear el robot y el objetivo.

Un espacio de trabajo con obstáculos estáticos y móviles, junto con las restricciones cinemáticas y dinámicas del robot imponen limitaciones que hacen que dichos controles óptimos en espacio libre no puedan aplicarse directamente. Sin embargo, el utilizar el modelo de *EVD* permite razonar fácilmente en términos de máxima velocidad y aceleración lineal o angular como controles extremos a ser aplicados para alinear el robot con el objetivo y/o evitar velocidades prohibidas que llevan a colisión, teniendo en cuenta las restricciones del robot. Estos son el tipo de controles de movimiento empleados en el planificador robocéntrico desarrollado en este trabajo. Como se describe en [Owen y Montano (2006)], este tipo de acciones de control resultan en trayectorias de tipo clotoide (sólo aceleración angular) y anti-clotoide (sólo aceleración lineal), que permiten al robot alinearse rápidamente en la dirección del objetivo o escapar de una colisión, bajo los criterios principales de seguridad del robot y evitación de colisión.

El planificador robocéntrico se ejecuta cíclicamente. La figura 3.2 muestra el *árbol de situaciones*, que se evalúa en cada ciclo determinando la estrategia de navegación. Las hojas representan las diferentes situaciones identificadas. Cada una de ellas tiene una estrategia de movimiento asociada que se establece aplicando jerárquicamente los criterios basados en las *variables de decisión* descritas en el anexo E.1. La jerarquía seguida para calcular el comando de movimiento para la estrategia correspondiente es diferente para cada situación; algunas veces, se seleccionan movimientos de velocidad alta, y otras, se opta por movimientos más seguros.

A continuación se describen las estrategias de navegación para las diferentes situaciones en el *árbol de situaciones*, mostrando los detalles en el anexo E.2.

4.1.1. MovimientoLibre

La situación *MovimientoLibre* corresponde al caso en el que no hay obstáculos móviles perceptibles por el sensor del robot o los que hay no suponen una amenaza. Se trata de la situación más sencilla y sirve para explicar la estrategia básica, que consiste en aplicar un criterio de alineación con el objetivo final, el *goal*.

En la figura 4.1 se muestra una secuencia de imágenes del *ET* durante la ejecución de la estrategia. Se puede ver cómo el robot gira para orientarse primero en la dirección del *goal*. Entonces, aumenta su velocidad para alcanzar el *goal* en el menor tiempo posible.

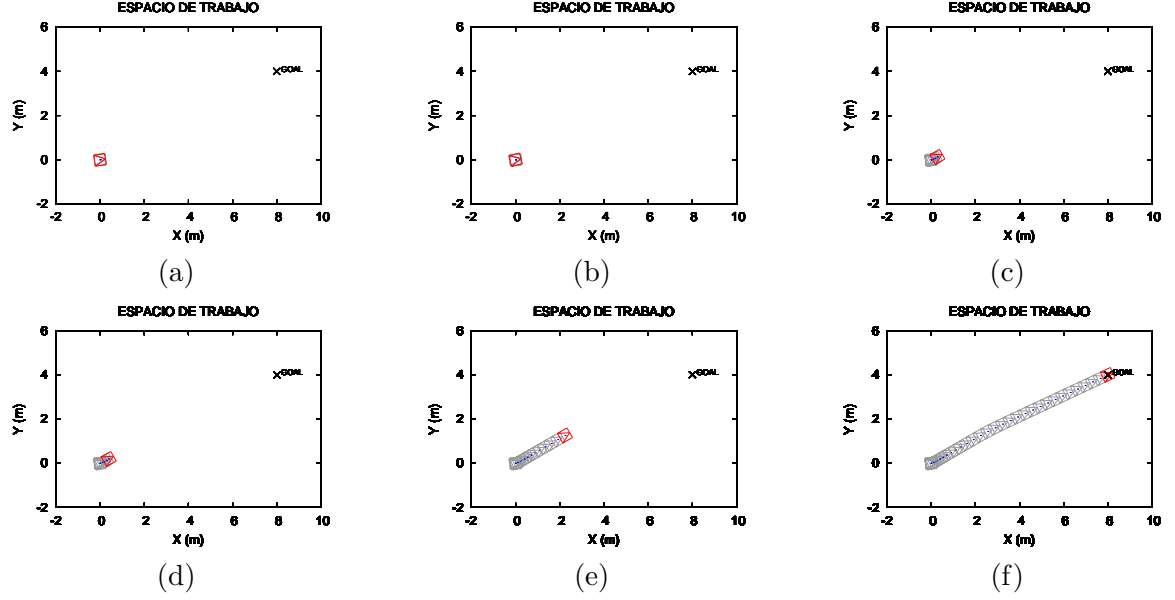


Figura 4.1: Situación *MovimientoLibre*. Evolución del *ET* durante la ejecución de la estrategia.

4.1.2. Pasar Por Delante

Implementa la estrategia *RobotAntes* de la figura 3.1. Bajo esta cabecera se hace referencia a todas aquellas estrategias de navegación que llevan al robot a pasar por delante del obstáculo.

Uno de los principales criterios de optimización utilizados en la planificación de movimientos es minimizar el tiempo en alcanzar el objetivo final. En espacio libre, esto conlleva aplicar comandos de velocidad y aceleración máximos, como ya se ha mencionado antes. Sin embargo, no siempre es posible en escenarios con obstáculos, estáticos o dinámicos. Dependiendo de las velocidades libres que pueden llegar a alcanzarse, se pueden distinguir dos situaciones distintas: *PasarPrimero* o *PasarAlineado*.

PasarPrimero

Representa una situación en la que el obstáculo no está muy cerca del robot y éste puede alcanzar velocidades lineales altas que le permiten pasar por delante del obstáculo sin peligro de colisión. La figura 4.2 contiene una secuencia de imágenes del *ET* de esta situación.

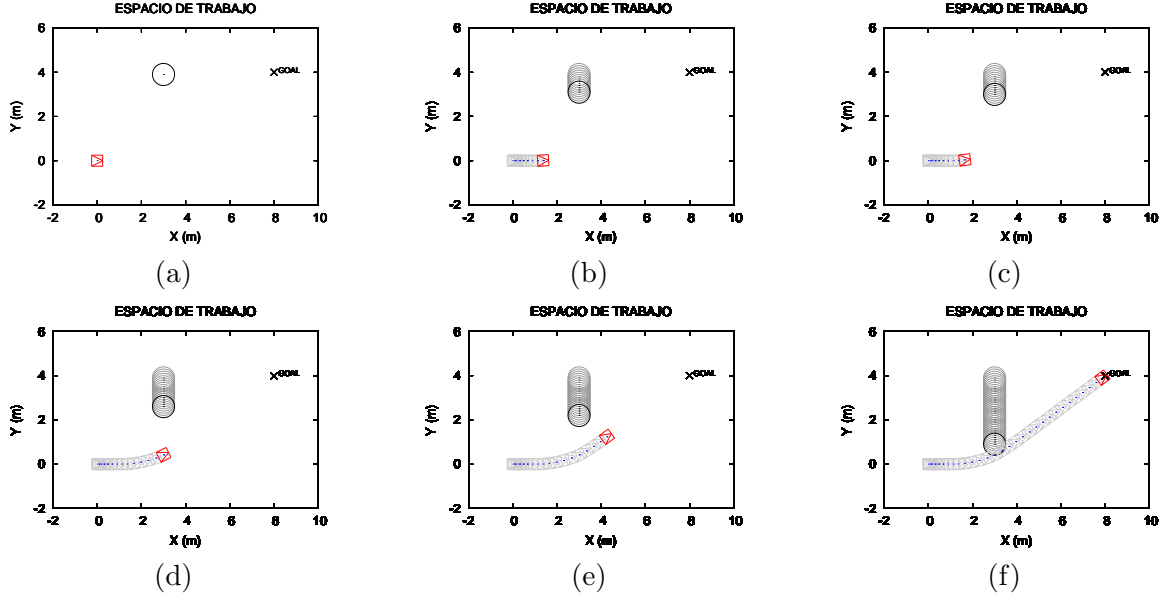


Figura 4.2: Situación *PasarPrimero*. Evolución del *ET* durante la ejecución de la estrategia.

PasarAlineado

En esta situación, el robot puede pasar por delante del objeto móvil ejecutando movimientos en la misma dirección que el obstáculo y a velocidad lineal alta. El *ET* en diferentes instantes durante la ejecución de la estrategia se muestra en la figura 4.3.

Cabe resaltar la importancia de la situación *PasarAlineado* dado que puede ser la única forma de escapar a una colisión con el objeto móvil.

4.1.3. Pasar Por Detrás

Implementa la estrategia *RobotDespues* mostrada en la figura 3.1. Con este nombre se incluyen todas aquellas estrategias de navegación que conducen al robot a pasar por detrás del objeto móvil. En función del conjunto de velocidades libres alcanzable, se pueden aplicar dos acciones: *Frenar* o *EvitandoObjeto*.

Frenar

Esta situación corresponde al caso en el que el obstáculo está muy próximo al robot y éste debe frenar de inmediato para evitar la colisión. En la figura 4.4 se muestra el *ET* durante la maniobra. Como se observa en las imágenes, el robot aprovecha para orientarse en la dirección del *goal* mientras espera a que pase el objeto móvil.

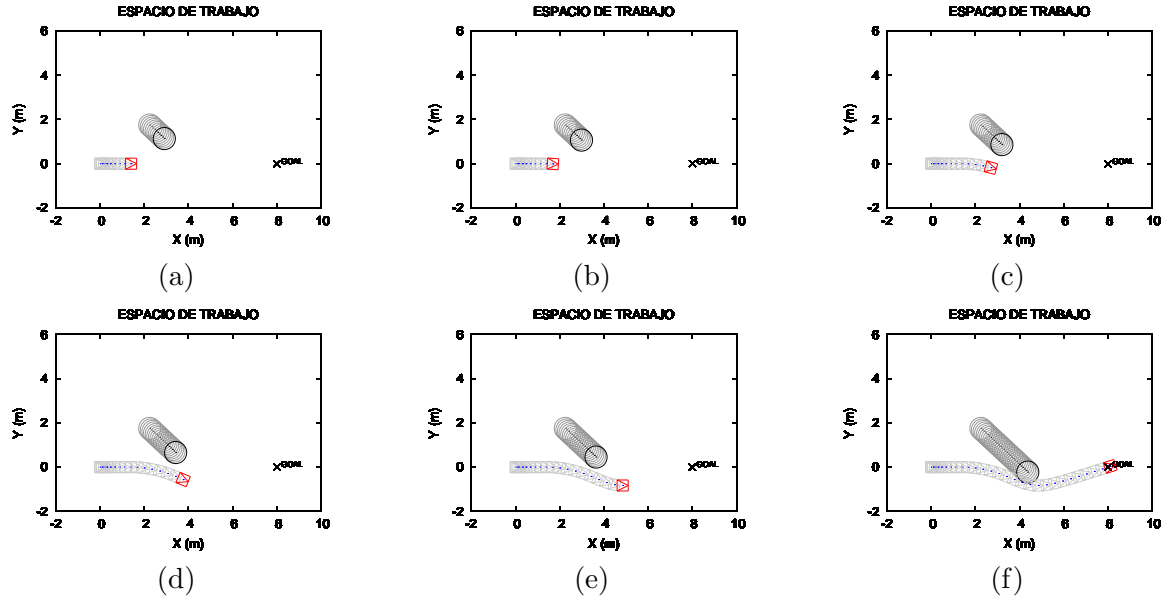


Figura 4.3: Situación *PasarAlineado*. Evolución del *ET* durante la ejecución de la estrategia.

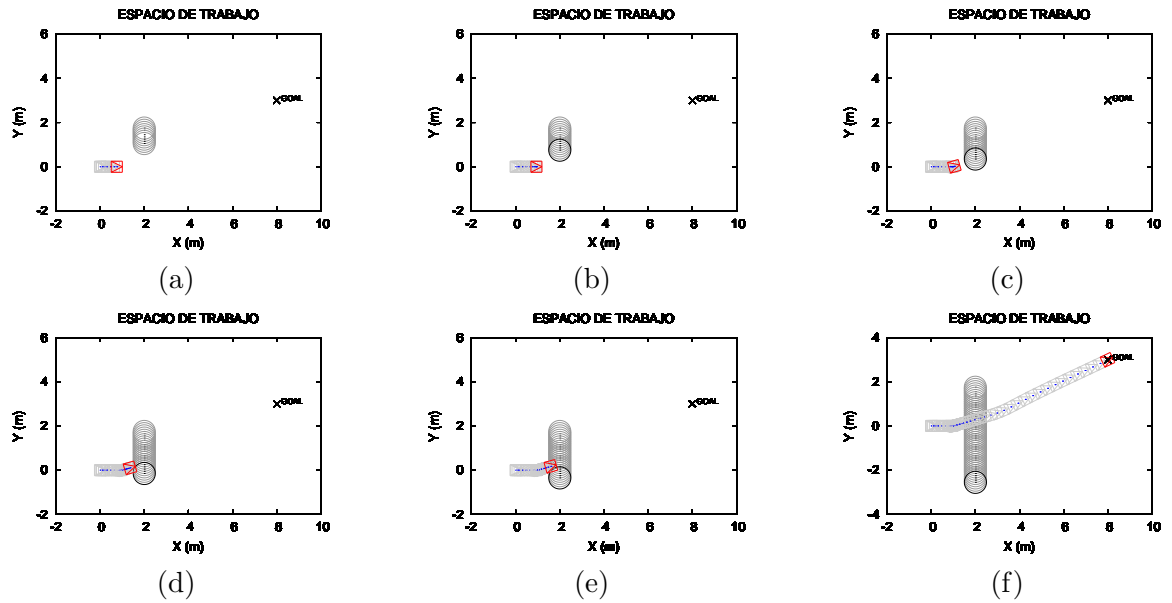


Figura 4.4: Situación *Frenar*. Evolución del *ET* durante la ejecución de la estrategia.

EvitandoObjeto

Se trata de una situación en la que el objeto móvil no se encuentra demasiado cerca del robot. Sin embargo, ya no es posible realizar una maniobra para adelantarlo, el robot lo evita

pasando por detrás. Al mismo tiempo, intenta orientarse hacia el *goal*. El *ET* se ilustra en la figura 4.5.

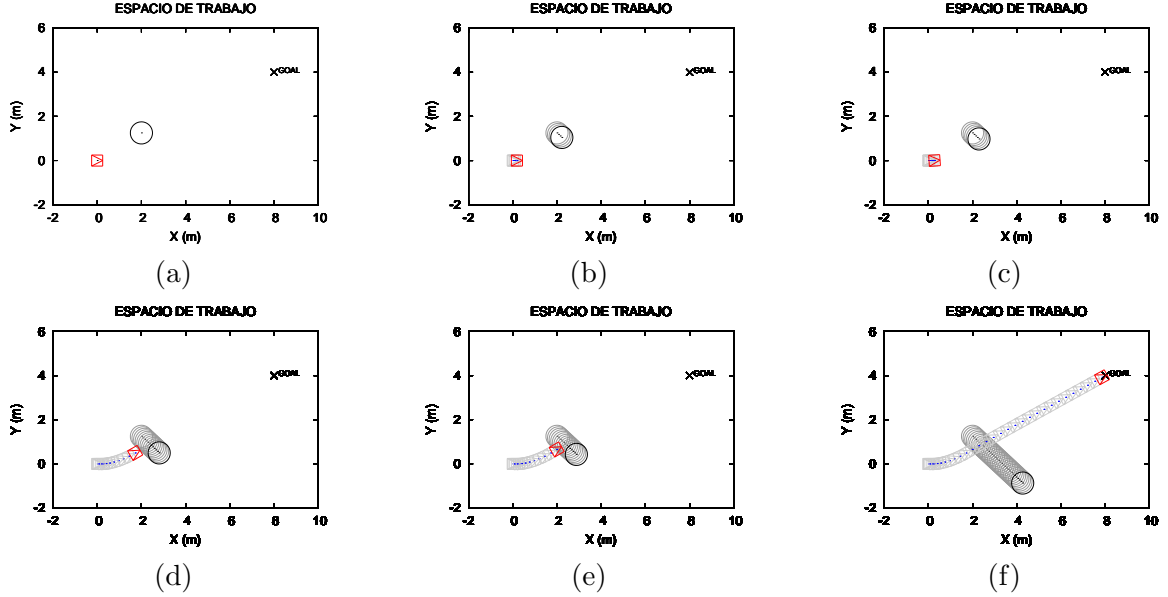


Figura 4.5: Situación *EvitandoObjeto*. Evolución del *ET* durante la ejecución de la estrategia.

4.1.4. Dentro de la Banda de Colision

Se refiere a situaciones en las que el robot está dentro de la *BC* del obstáculo. El robot tiene que ejecutar velocidades lineales altas para escapar de esta situación y avanzar hacia el *goal*, aunque no siempre será posible. Cuando un objeto aparece de repente en el escenario cerca del robot y con velocidad mayor que la del robot, se produce una situación de *CollisionSegura*, lo que significa una inevitable colisión con el objeto móvil. En este caso, no es posible reaccionar para evitar la colisión.

Las figuras 4.6 y 4.7 reflejan escenarios en estas dos situaciones distintas.

4.1.5. Varios obstáculos

Las situaciones explicadas en los apartados anteriores son también válidas para aquellos casos en los que hay más de un obstáculo en el escenario. Básicamente, se puede hacer dos distinciones sobre las situaciones: una, en la que se selecciona *PasarPrimero* para permitir al robot pasar por delante de varios obstáculos; y otra, en la que la situación sería la correspondiente respecto al obstáculo que representa mayor peligro para la navegación del robot.

La figura 4.8 corresponde al primer caso. El planificador selecciona la situación *PasarPrimero* conduciendo al robot a pasar por delante de los dos objetos. La maniobra seguida se

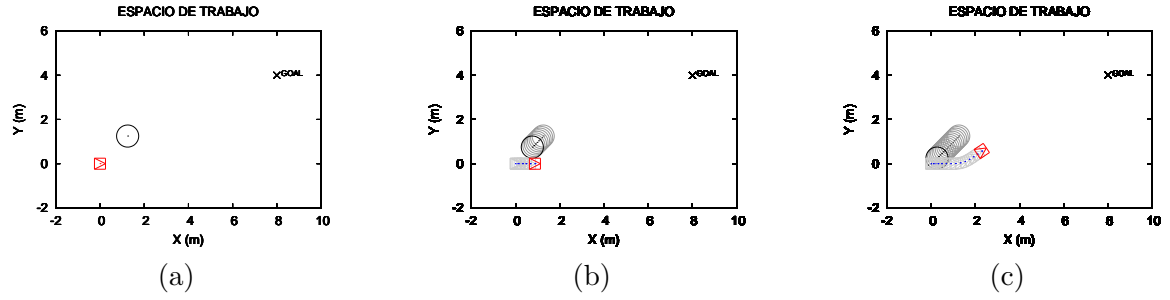


Figura 4.6: Dentro de la banda de colisión. Evolución del ET durante la ejecución de la estrategia. El robot escapa de la colisión.

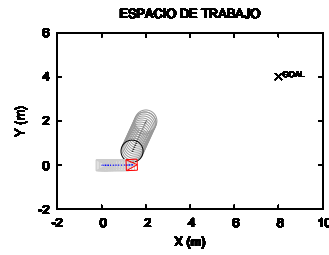


Figura 4.7: Situación *ColisionSegura*. Evolución del ET durante la ejecución de la estrategia. El robot no puede escapar a la colisión.

describe en el anexo E.2.5.

En cuanto al segundo caso, la figura 4.9 ilustra un escenario con dos objetos móviles. El robot frena dejando pasar al obstáculo más cercano. Posteriormente, evita el siguiente objeto pasando por delante de él. La secuencia de movimientos se detalla en el anexo E.2.5.

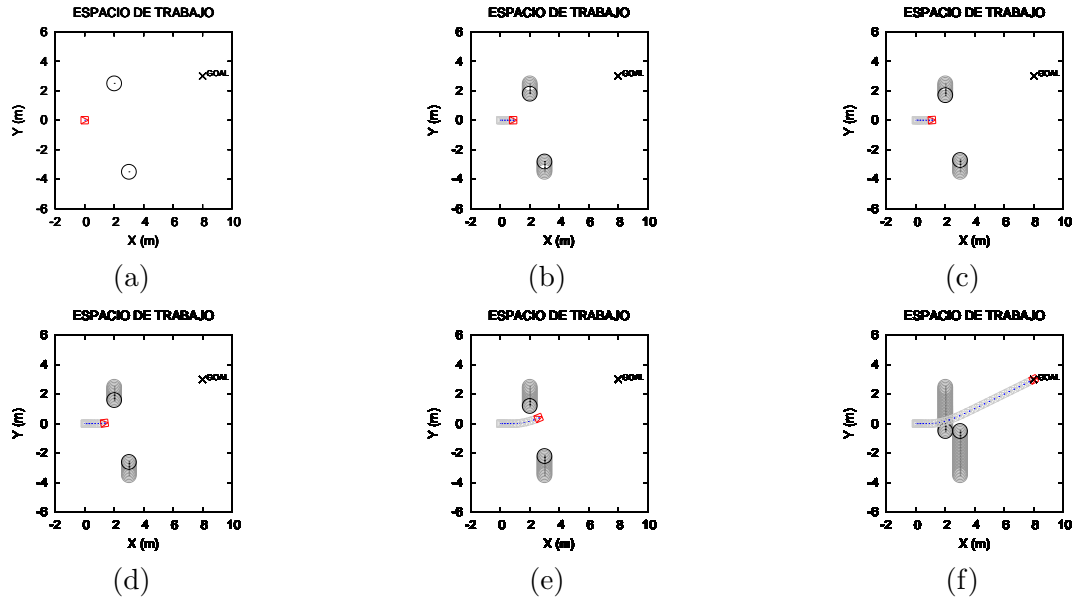


Figura 4.8: Situación *PasarPrimero*. Evolución del *ET* durante la ejecución de la estrategia.

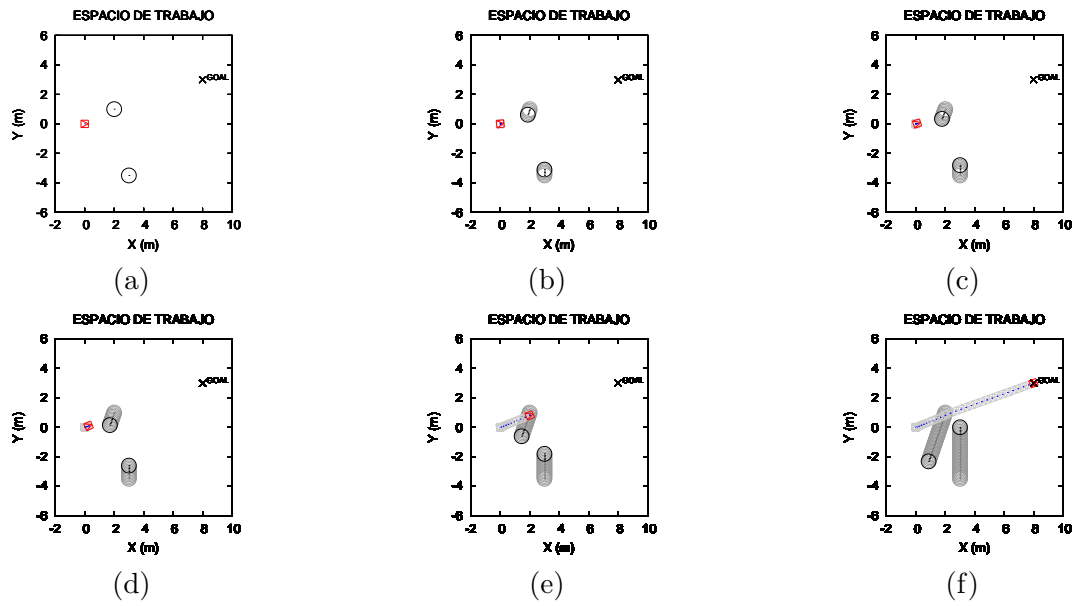


Figura 4.9: Situación con varios obstáculos. Evolución del *ET* durante la ejecución de la estrategia. La situaciones seleccionadas son *Frenar* y *PasarPrimero*.

Capítulo 5

Evaluación de la planificación de la navegación

El objetivo de este capítulo consiste en evaluar y validar las estrategias de navegación propuestas para entornos dinámicos, basadas en la descripción e implementación de las diferentes situaciones en el *EVD*.

Dado que la navegación en este tipo de entornos supone una tarea crítica para el robot, se han testado diferentes situaciones verificando la seguridad y eficiencia de las estrategias. Los resultados de simulación que se presentan en este capítulo así como los de experimentación, capítulo 6, pretenden ilustrar la funcionalidad y validez de las estrategias.

A lo largo del capítulo se muestra un conjunto de escenarios complejos por los que el robot debe navegar. Estos escenarios se diseñaron inicialmente para C/C++, para realizar una primera versión del sistema y los primeros ajustes de los parámetros y más tarde, sobre [Player/Stage], para obtener una visión más realista del sistema. Los resultados que se presentan pertenecen a las pruebas con Player/Stage.

Las simulaciones se realizaron para verificar que las estrategias de navegación satisfacen el objetivo principal de este proyecto: *conducir un robot móvil de forma segura en presencia de objetos estáticos y dinámicos*.

5.1. Caos

Con el primer experimento, denominado *caos*, se intenta minar la capacidad de navegación del robot en un escenario impredecible y no estructurado donde varios obstáculos se mueven a su alrededor restringiendo sus movimientos de forma considerable. Cualquier estrategia pasiva sería extremadamente peligrosa, por lo que el robot debe iniciar la navegación tan pronto como sea posible y aplicar una estrategia segura para evitar colisiones mientras se dirige hacia el *goal*.

Tal y como se observa en la imagen 5.3a hay tres obstáculos móviles que representan mayor peligro para la navegación segura del robot (objetos *obj2*, *obj4* y *obj5*). El sistema sólo tiene en cuenta aquellos obstáculos móviles que están muy cerca del robot y suponen una amenaza para la navegación, tratando posteriormente los obstáculos más lejanos. Así, la carga computacional del sistema se reduce sin comprometer la seguridad.

La figura 5.3 ilustra varias imágenes del *ET* durante la simulación. El resultado final de la maniobra se resume como un giro hacia la derecha que hace que el robot pase entre los objetos *obj4* y *obj5*. La figura 5.1a muestra el perfil de velocidad lineal, se corresponde a un movimiento uniformemente acelerado hasta que se alcanza la velocidad lineal máxima. Es un movimiento suave y libre de oscilación a pesar de los obstáculos que le rodean. El perfil de velocidad angular de la figura 5.1b refleja también movimientos suaves, sin cambios bruscos. En el perfil de situaciones de la figura 5.2 se puede ver que *PasarPrimero* y *EvitandoObjeto* se aplican durante la simulación para maniobrar.

En el anexo F.1 se ofrece una descripción más detallada de los movimientos seguidos así como una secuencia más completa de imágenes durante la simulación.

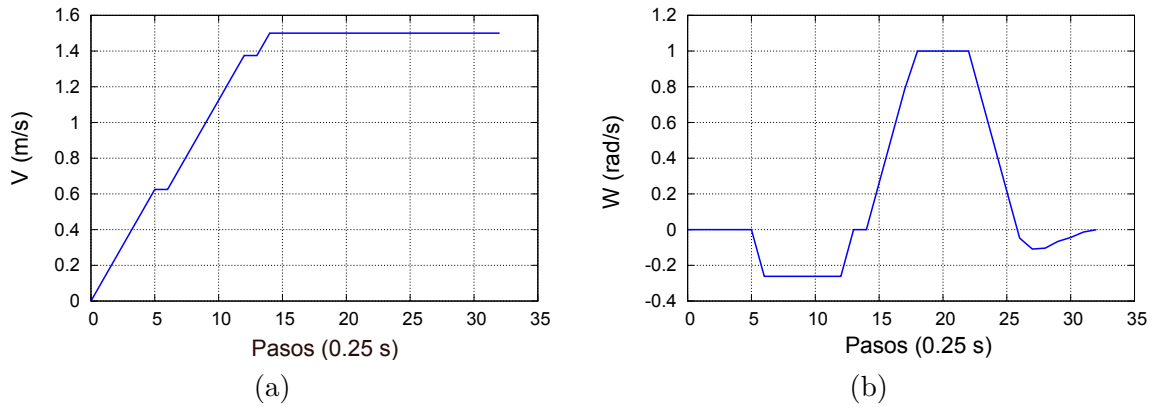


Figura 5.1: Perfiles de velocidad lineal (a) y angular (b) del robot en la simulación *caos*.

5.2. Carriles

En este experimento, *carriles*, el robot se enfrenta a un escenario más estructurado en el que los objetos dinámicos describen movimientos rectilíneos, similares al movimiento de los coches cuando circulan por una autovía, y el robot tiene que cruzar por en medio manteniendo su integridad.

La finalidad de la simulación es validar el comportamiento y seguridad del sistema en un escenario donde las direcciones de los objetos móviles están restringidas y el robot debe cruzar tan rápido como le sea posible.

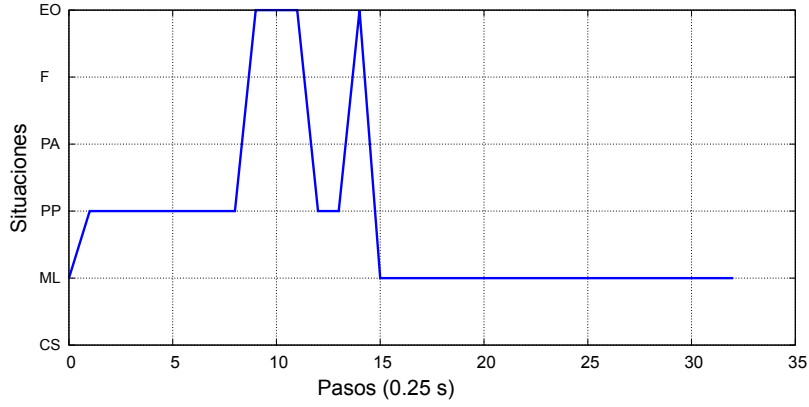


Figura 5.2: Perfil de situaciones de la simulación *caos*.

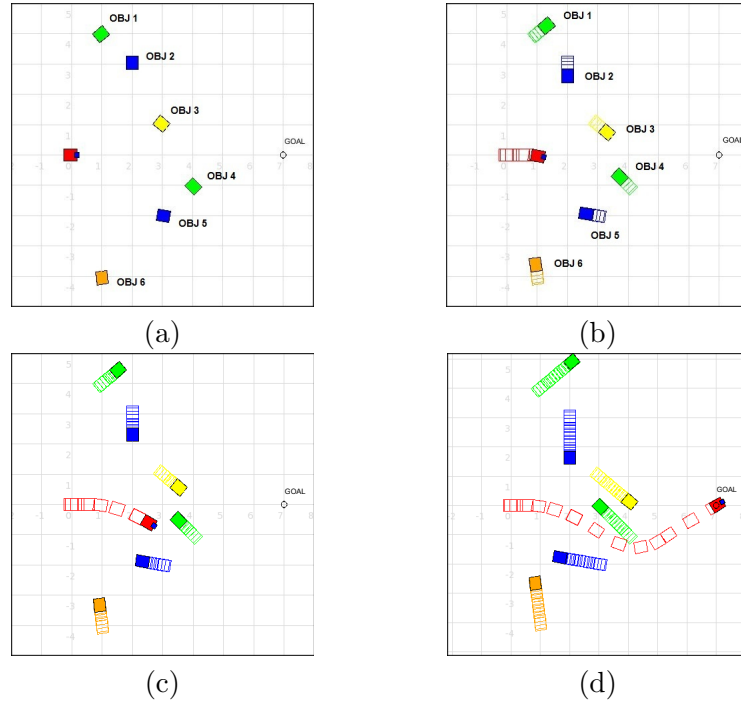


Figura 5.3: Evolución del *ET* a lo largo de la simulación *caos*.

En la figura 5.6 se muestran imágenes del *ET* en diferentes instantes de la simulación. El planificador mantiene al robot a velocidades bajas mientras los obstáculos *obj1*, *obj2* y *obj3* están pasando. A continuación, se resuelve la navegación pasando por detrás del *obj4* con *EvitandoObjeto*. Finalmente, el robot pasa por delante del objeto *obj5* para alcanzar el *goal*.

Los perfiles de velocidad lineal y angular se muestran en la figura 5.4. De nuevo, se puede

ver que los perfiles son suaves y libres de oscilación, manteniendo la velocidad o acelerando cuando es necesario. En la figura 5.5 se muestra la secuencia de situaciones seleccionada por el planificador: *Frenar*, *EvitandoObjeto* y *PasarPrimero*.

En el anexo F.2 se ofrecen más detalle sobre la simulación.

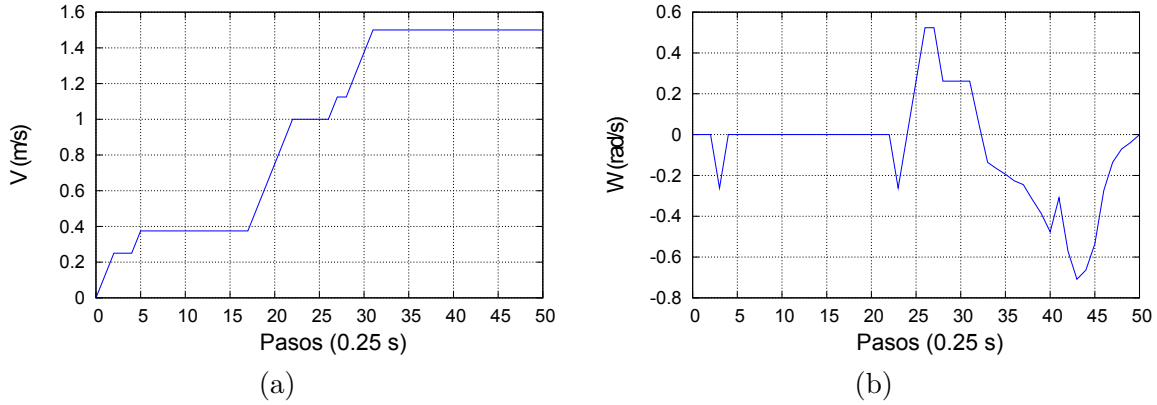


Figura 5.4: Perfiles de velocidad lineal (a) y angular (b) en *carriles*.

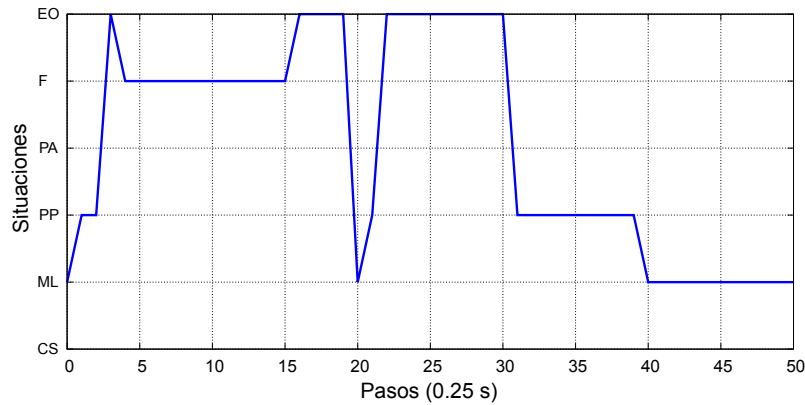


Figura 5.5: Perfil de situaciones en *carriles*.

5.3. Adelantamiento

En el último experimento, *adelantamiento*, el robot se comporta de forma similar a un vehículo que circula por una vía e intenta realizar un adelantamiento a otro vehículo más lento. Se muestra cómo la técnica general propuesta puede emplearse para este tipo de aplicaciones.

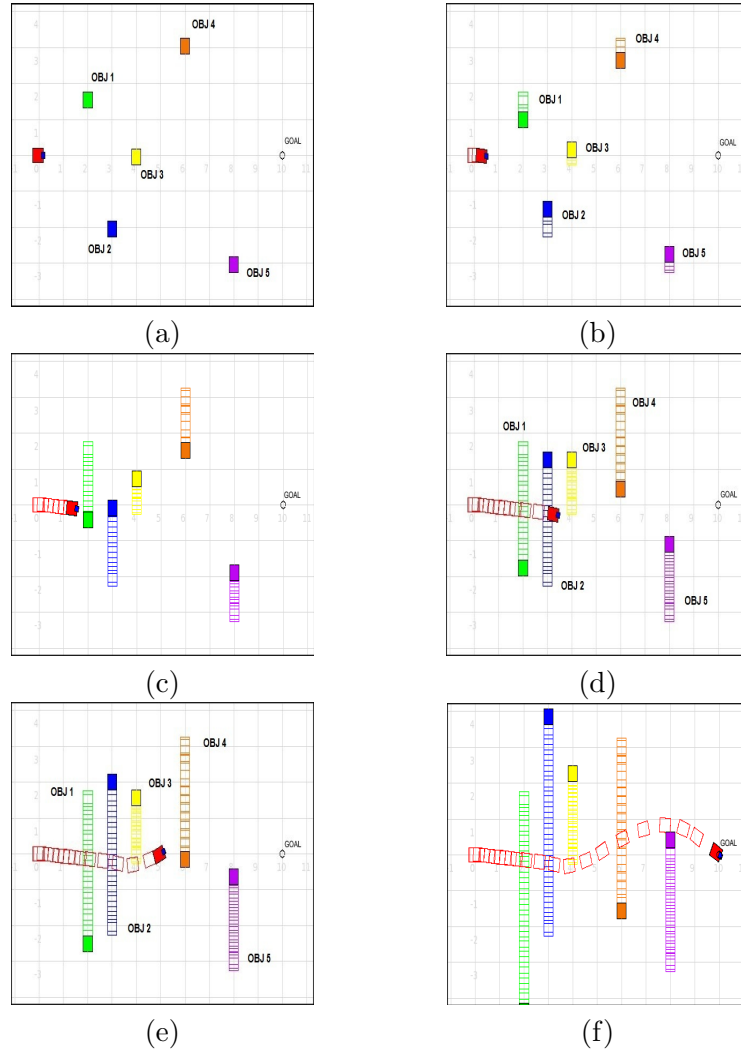


Figura 5.6: Evolución del ET a lo largo de la simulación *carriles*.

El objetivo de la simulación es validar el comportamiento y seguridad del sistema en un escenario de este tipo, en el que se necesitan maniobras para volver al carril de la derecha. Además, aparecen objetos estáticos virtuales, se trata de las líneas que marcan los dos lados de la carretera.

El ET y el EVD a lo largo de la simulación queda reflejado en la figura 5.9. Como la distancia entre el robot y el obstáculo *obj1* es grande, el robot puede navegar libremente por su carril aumentando su velocidad lineal. Cuando está cerca del *obj1*, puesto que su velocidad es baja y el *obj2* que se aproxima por el carril contrario está todavía lejos, el robot inicia la maniobra de adelantamiento para evitar al *obj1* por la izquierda a máxima velocidad lineal. Una vez lo adelanta, el robot vuelve a su carril completando la maniobra satisfactoriamente.

La figura 5.7 muestra los perfiles de velocidad lineal y angular. Toda la maniobra se realiza a máxima velocidad lineal. En el perfil de situaciones de la figura 5.8 se ve cómo las situaciones seleccionadas por el planificador son *MovimientoLibre* y *EvitandoObjeto*, dependiendo de si el robot está cerca o lejos de los obstáculos.

El apéndice F.3 recoge una explicación más amplia con una secuencia de imágenes mayor durante la simulación.

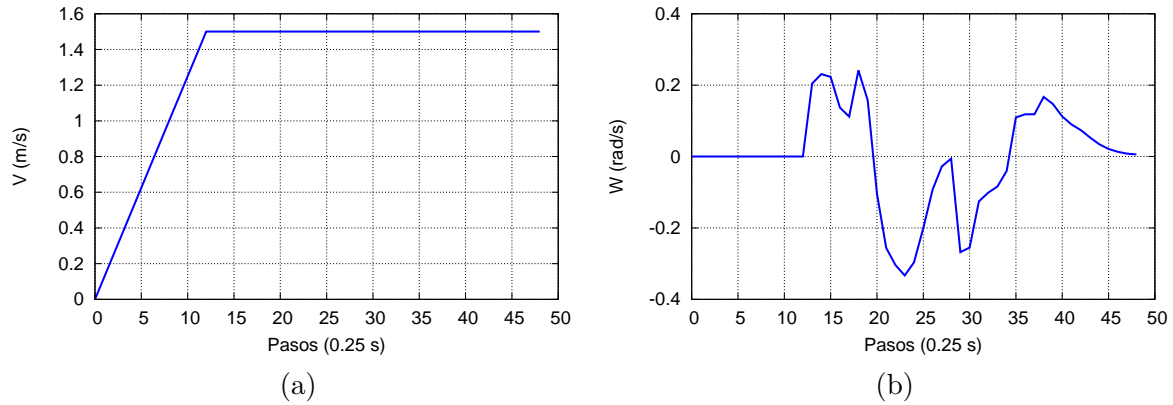


Figura 5.7: Perfiles de velocidad lineal (a) y angular (b) del robot durante la simulación *adelantamiento*.

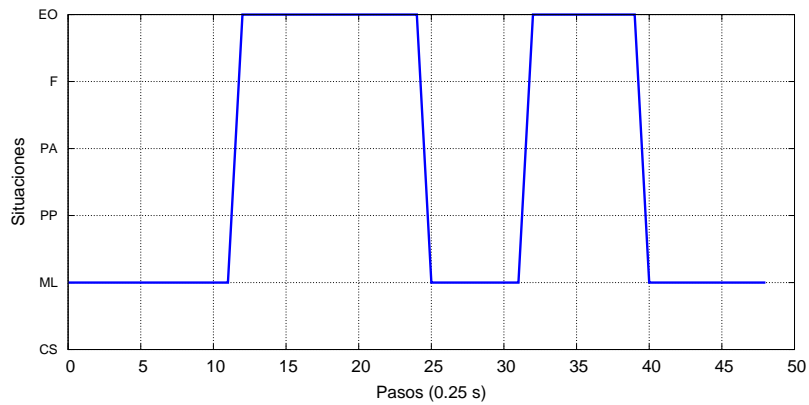


Figura 5.8: Perfil de situaciones del *adelantamiento*

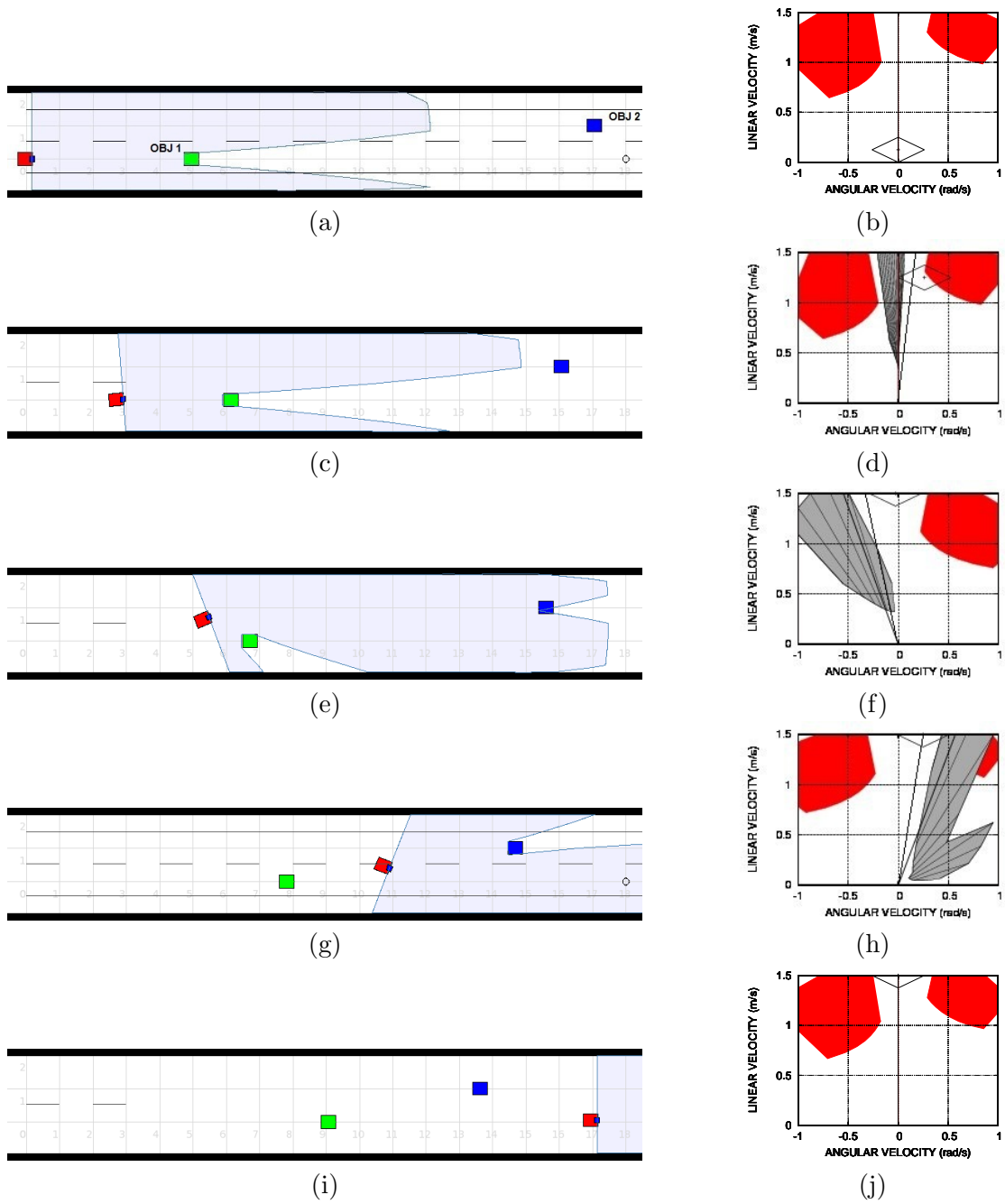


Figura 5.9: Evolución del *adelantamiento*. La columna de la izquierda refleja *ET* y la de la derecha, el *EVD* correspondiente.

Capítulo 6

Experimentación real

En el capítulo anterior se mostraba la funcionalidad del sistema de navegación mediante una serie de simulaciones cuyos resultados fueron satisfactorios. Sin embargo, también se puso a prueba el sistema en un entorno real para poder validar su funcionamiento y considerar e identificar posibles limitaciones no perceptibles desde simulación. Las estrategias de navegación se han evaluado en dos escenarios reales.

6.1. Preparación de los experimentos

Las estrategias se han testado sobre un robot Pioneer del grupo de Robótica. Se trata de un robot diferencial [Asensio y Montano] equipado con un láser 2-D Sick LMS-200 en la parte frontal y un procesador Intel Centrino duo a 1.6GHz. El tiempo de cálculo para las estrategias de navegación fue alrededor de 200 ms, aproximadamente, el tiempo en el que una nueva medida del sensor está disponible proporcionando información actualizada del entorno. El campo de visión del láser es de 180° con resolución angular de $0,5^\circ$ y un alcance máximo de 8 metros. Se fijó la velocidad máxima de traslación a 0.4 m/s y la de rotación a 0.3 rad/s por dos razones: la primera, porque los experimentos se realizan en un entorno de interior, y así se tiene mayor seguridad; la segunda, porque estas velocidades pueden ser simultáneamente alcanzadas por el robot diferencial ya que sus velocidades máximas reales son mayores que dichos valores, permitiendo ajustarse al cuadrado de la ventana dinámica utilizado en el método.

La técnica empleada para seguir el movimiento de los obstáculos se basa en el *Filtro de Kalman Extendido* (apéndice C.1), en el que el vector de estado incluye la posición y velocidad de los obstáculos seguidos. El mapa de los objetos estáticos y dinámicos se obtiene a través del método desarrollado en [Montesano y otros] para modelar un entorno dinámico. Empleando esta técnica, se identifican las partes estáticas y dinámicas del escenario, por lo que la información se utiliza directamente para ser mapeada en el modelo de *EVD*.

6.2. Experimento 1: navegación por un hall

En el primer experimento, el robot intenta alcanzar el objetivo (*goal*) en un escenario tipo hall, similar al de la simulación *caos*, donde tres obstáculos se mueven libremente restringiendo la navegación del robot. El experimento consiste en alcanzar el *goal* evitando colisionar con los obstáculos. En la figura 6.3 se muestran varias capturas de la prueba real. En el apéndice G se describe en detalle los movimientos realizados por el robot.

Las maniobras seleccionadas por el planificador de navegación mantienen la máxima velocidad lineal durante todo el experimento mientras el robot evita los obstáculos y se dirige hacia el *goal*. El planificador selecciona *PasarPrimero* para pasar secuencialmente los objetos *obj1* y *obj2*. Una vez superado el *obj1*, se escoge *EvitandoObjeto* para terminar de evitar el *obj2* y pasar a *MovimientoLibre*. A continuación, se detecta *obj3* y el robot debe girar para evitarlo a máxima velocidad lineal y angular, ejecutando *EvitandoObjeto* hasta que se evita completamente y se alcanza la situación de *MovimientoLibre*. La secuencia de situaciones queda reflejada en la figura 6.2.

La navegación es segura a máxima velocidad, y la situación *ColisionSegura* no se alcanza nunca. Los objetos estáticos y móviles fueron detectados y seguidos correctamente, y el movimiento fue libre de fuertes oscilaciones, tal y como se puede ver en las figuras 6.1a,b.

En la gráfica 6.2b se muestra la trayectoria descrita por el robot durante el experimento (puntos en verde) además de los puntos proporcionados por el láser, clasificados en estáticos (azules) y dinámicos (rojos). Se puede observar también la trayectoria descrita por los obstáculos.

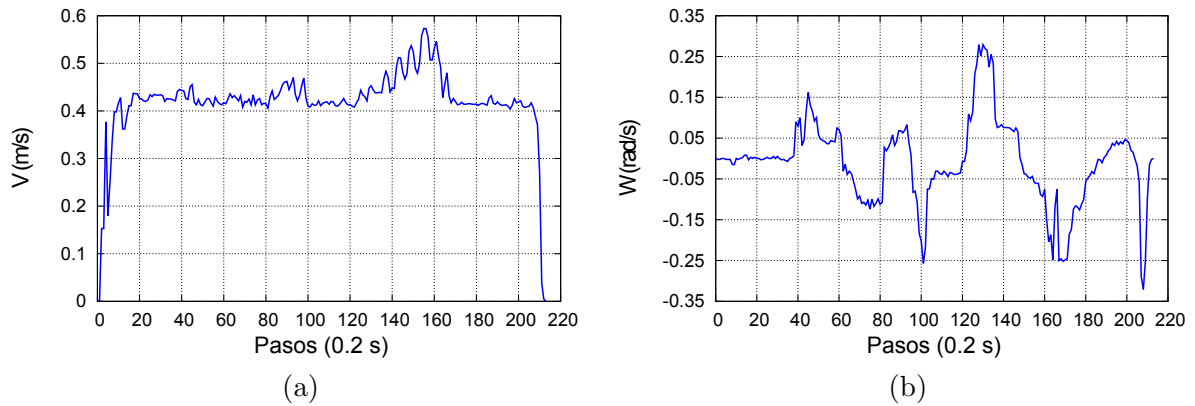


Figura 6.1: Perfiles de velocidad lineal (a) y angular (b) durante el experimento 1.

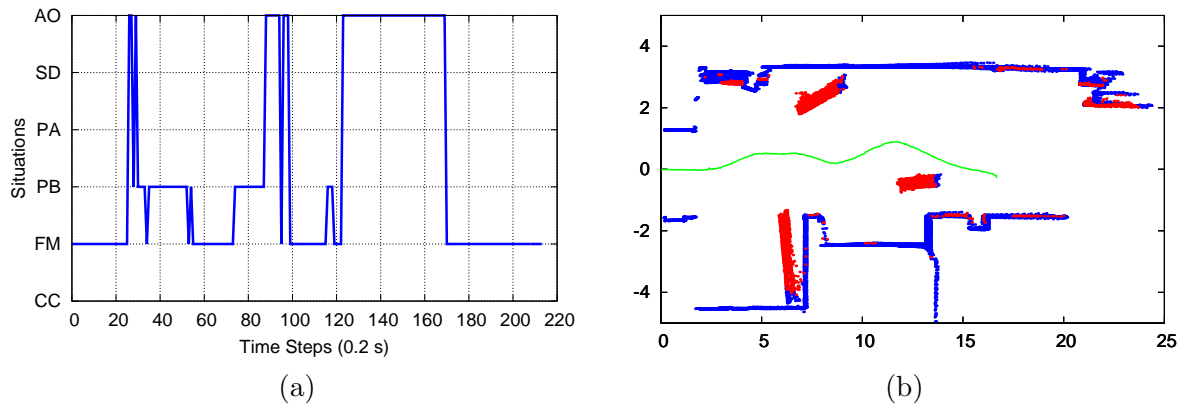


Figura 6.2: (a) Perfil de situaciones durante el experimento 1, (b) trayectoria del robot (verde) y entorno percibido en el experimento 1.

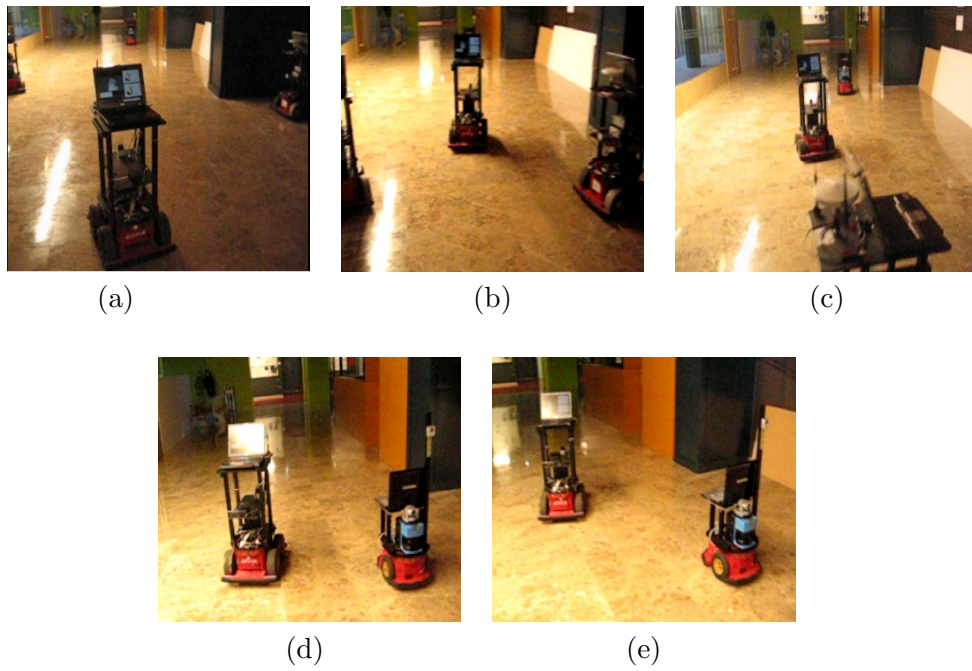


Figura 6.3: Diferentes instantes durante el experimento 1.

6.3. Experimento 2: navegación por un pasillo

El escenario donde se realiza este experimento es un pasillo estrecho, imitando la simulación del *adelantamiento*, donde se reduce la capacidad de navegación del robot.

La finalidad del experimento es que el robot alcance el *goal* mientras evita la colisión con otros obstáculos en un espacio de movimiento restringido. Como en el caso de la simulación, el espacio se divide virtualmente en dos carriles que utilizará el robot para navegar, circulando por el carril derecho siempre que sea posible. La figura 6.6 muestra varias capturas del vídeo del experimento real. En el apéndice G.2 se detalla la secuencia de movimientos realizados por el robot.

El planificador selecciona *Frenar* debido a que los obstáculos *obj1* y *obj2* impiden su navegación. Una vez que el *obj2* ha pasado, el robot puede adelantar al *obj1* pasando a *EvitandoObjeto*. Más adelante, se detecta un tercer objeto *obj3* que no interfiere en la navegación del robot, continuando con la situación anterior hasta que pasa a ser *MovimientoLibre* cuando desaparece de la visibilidad del robot. Finalmente, el planificador escoge *EvitandoObjeto* para evitar el *obj4* y alcanzar el *goal*. En la figura 6.5 se muestra el conjunto de situaciones durante la simulación.

La navegación se realiza de forma segura sin encontrar ninguna situación de tipo *ColisionSegura*. En la figura 6.4 se ve que el robot navega a máxima velocidad la mayor parte del tiempo, excepto cuando tiene que frenar en la primera parte porque hay poco espacio para maniobrar, y que los movimientos realizados fueron suaves.

La figura 6.5b muestra la trayectoria seguida por el robot durante el experimento y la información percibida por el láser. Se puede observar cómo, al final de la trayectoria, los puntos aparecen un poco desviados. Esto se debe al error de odometría que el robot va acumulando durante la navegación. A pesar de que se va corrigiendo a medida que el robot avanza, es inevitable que siga existiendo cierto error.

Todas las situaciones definidas en el árbol de situación aparecen en las simulaciones y experimentos, excepto *ColisionSegura* y *PasarAlineado*. La primera nunca se selecciona porque ningún objeto entró en el escenario de repente, todos ellos fueron detectados dentro del campo de visión del sensor y se seleccionó el comando apropiado para la situación. *PasarAlineado* no fue detectada porque ésta se produce cuando un objeto se aproxima rápidamente hacia el robot y está muy cercano a la banda de colisión. Esta situación está también asociada a cambios repentinos en el escenario, los cuáles no suceden en los experimentos.

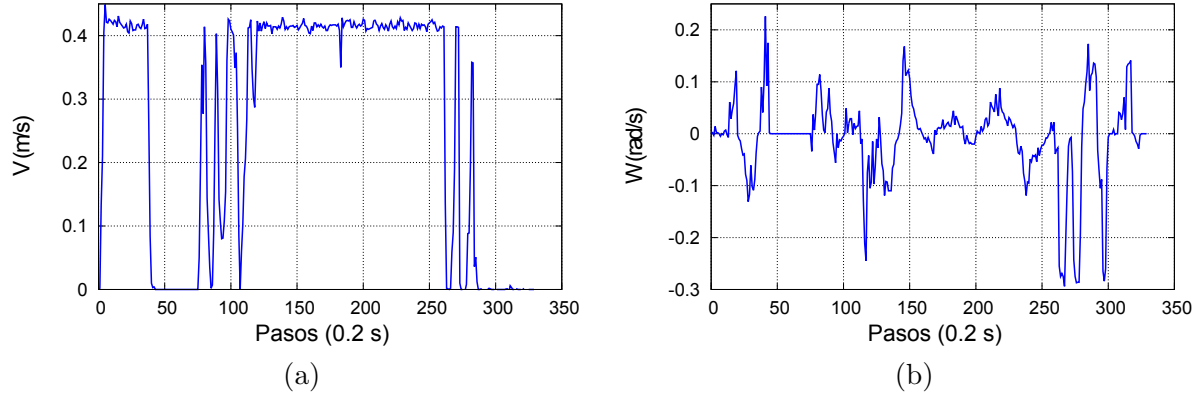


Figura 6.4: Perfiles de velocidad lineal (a) y angular (b) del robot durante el experimento 2.

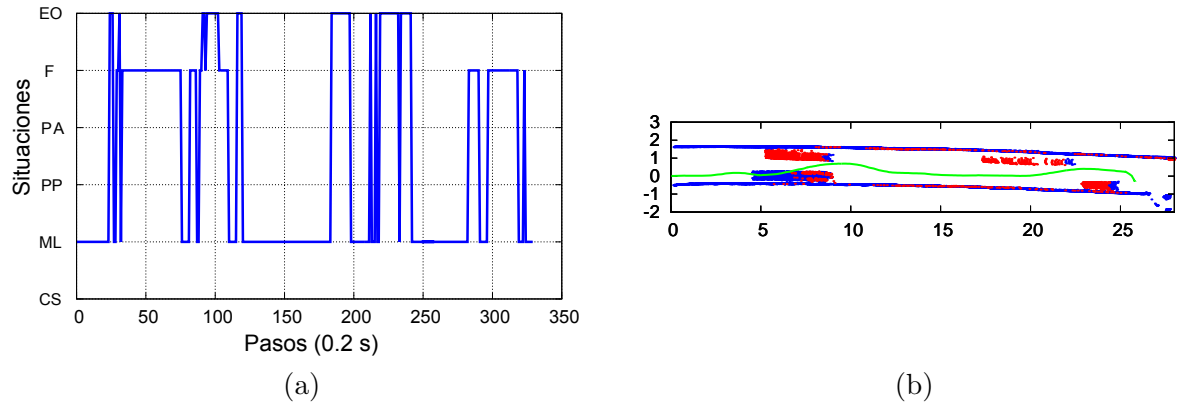


Figura 6.5: (a) Perfil de situaciones durante el experimento 2, (b) trayectoria del robot en el experimento 2.



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)



(k)



(l)

Figura 6.6: Capturas del video del experimento 2.

Capítulo 7

Conclusiones y trabajo futuro

En este capítulo se presentan algunas consideraciones sobre la técnica implementada como resultado de todo el trabajo llevado a cabo durante la realización de este PFC, así como algunas limitaciones y problemas encontrados y el planteamiento de líneas futuras de desarrollo.

7.1. Conclusiones

El trabajo realizado supone una nueva aportación al campo de la navegación segura de robots en entornos dinámicos. Los objetivos marcados inicialmente en el proyecto y los nuevos que han ido surgiendo durante su realización se han cumplido satisfactoriamente.

En este PFC se ha presentado el diseño e implementación de una técnica de planificación de navegación robocéntrica en tiempo real para entornos dinámicos que calcula los comandos de movimiento directamente en el espacio de velocidades.

El método maneja las restricciones cinemáticas y dinámicas explícitamente sobre el modelo y calcula comandos de movimiento aplicables a vehículos reales sin aproximaciones ya que utiliza la ventana dinámica del robot. La técnica se basa en identificar una situación de entre un conjunto finito de situaciones calculadas a partir de la información percibida por un sensor láser y que es mapeada en el espacio de velocidades. La navegación segura depende de varias variables relacionadas con la posición y velocidad relativas de los objetos y el robot, haciendo de la toma de decisiones una tarea compleja. La técnica propuesta maneja las variables relacionadas y en base a ellas, calcula un comando de movimiento seguro en cada periodo de tiempo.

Las simulaciones y los experimentos reales han demostrado el comportamiento del robot en diferentes condiciones y escenarios. Todas las situaciones definidas en el método pueden ser alcanzadas en algún momento, conduciendo siempre a movimientos seguros y cuando es posible, a movimientos rápidos. El método saca provecho de todas las posibilidades de maniobra seleccionando las velocidades en el *EVD*, obteniendo en cada momento el comando más

adecuado. Durante los experimentos no se detectó ninguna situación de inevitable colisión ni de pasar alineado.

El conjunto de situaciones y estrategias obtenido proporciona un sistema de planificación fiable para navegar de forma segura en escenarios desconocidos y dinámicos.

7.2. Limitaciones y problemas encontrados

El método propuesto resuelve el problema de la navegación en entornos dinámicos. Sin embargo, aparecen limitaciones que tienen que ver con el escenario donde se aplica la técnica, en este caso, un pasillo en forma de U. Se realizó una prueba situando el robot de frente a la pared y el *goal* justo detrás de él. El resultado que proporcionó el planificador fue una consecución de comandos de velocidad angular positiva y negativa que llevaban al robot a rotar a izquierda y derecha pero sin avanzar, buscando la zona de mayor espacio de velocidad libre en cada periodo de tiempo, sin llegar a resolver el problema. Ante este caso, sería conveniente disponer de algún criterio adicional como *variable de decisión* que permitiera seguir con la trayectoria una vez se ha iniciado.

En un entorno de tipo pasillo, como el del *adelantamiento* y el experimento 2, el hecho de que el campo de visión del láser sea de 180° hace que los laterales del *EVD*, es decir, las zonas relativas a las máximas velocidades angulares, aparezcan libres de colisión tras mapear los objetos estáticos, tal y como se refleja en las figuras 5.9b y G.3b. Razonar sobre este *EVD* lleva a pensar que la selección de dichos comandos de velocidad conduciría al robot a evitar los obstáculos estáticos. Sin embargo, un movimiento hacia alguna de esas zonas lleva a que el espacio ocupado en el *EVD* sea mayor. Esto hace que se pierda tiempo en considerar zonas de velocidad y por tanto, trayectorias, que realmente no conducen al *goal*.

Además, conviene tener en cuenta el tipo de escenario en el que se empleará el método y seleccionar un horizonte de tiempo adecuado que permita representar la información relevante del entorno pero que no imposibilite el encontrar una solución. Nos enfrentamos a esta restricción en el segundo experimento. Inicialmente, se empleó el mismo horizonte de tiempo que para el primer experimento, 1.5 m/s, pero se observó que al modelar un escenario tan reducido en el *EVD* la zona de velocidades superiores se cerraba tan sólo con la información de las paredes del pasillo, sin ningún otro obstáculo. Por lo tanto, hubo que adaptar dicho parámetro para permitir encontrar una solución a la navegación pero que no fuera demasiado pequeño y limitara su visibilidad representando tarde los objetos en el *EVD* o no lo suficientemente pronto como para reaccionar y poder evitar una colisión, poniendo en peligro el robot.

Estos problemas no se encontraron en las pruebas con sólo dinámicos ni en el experimento 1, donde el espacio de navegación era mayor.

A nivel personal, uno de los principales problemas al que tuve que hacer frente durante la realización del PFC fue familiarizarme con conceptos desconocidos hasta entonces, como son

el funcionamiento de un robot diferencial y las transformaciones entre el sistema global (el mundo) y el del robot para poder tratar la información desde el punto de vista robocéntrico. El entender el concepto general detrás del software disponible para la localización, identificación y seguimiento de los obstáculos así como el estudio del *FKE* para proporcionar unos parámetros iniciales adecuados y un centroide como nueva medida para el filtro, llevó su tiempo.

En la etapa de experimentación, hubo que realizar varias pruebas previas para conocer la ventana dinámica del robot y establecer los valores de velocidad máximos para permitir al robot responder y ejecutar los comandos calculados por el planificador.

7.3. Trabajo futuro

Tras analizar los resultados y limitaciones encontradas, se plantean posibles mejoras o aplicaciones como líneas de trabajo futuro. Se resumen a continuación:

- Implementación de un planificador global que haga uso del método de planificación local desarrollado. Así, el planificador global marcaría una serie de subobjetivos a cumplir por el robot para alcanzar el objetivo final. En la navegación hacia esos subobjetivos actuaría la técnica implementada.

Muchos desarrollos en Robótica respecto a la navegación de robots en entornos dinámicos implementan un árbol de tipo A^* para hacer la búsqueda de la mejor solución posible durante el periodo de tiempo disponible. Se podría pensar en este tipo de árbol para el planificador global, que realizara una búsqueda de la mejor estrategia a ejecutar empleando el método presentado en este PFC como heurística para podar ciertas ramas en función de la situación identificada, acelerando así el proceso de búsqueda hacia el mejor movimiento.

El planificador global podría calcular también una trayectoria inicial hacia el *goal* y con conocimiento previo del entorno podría discernir aquellas zonas que llevarían al robot a situaciones de atrapamiento. Con la técnica implementada se modificaría dicha ruta inicial para permitir al robot actuar de forma reactiva ante la presencia de obstáculos.

- Empleo de la variable del tiempo como tercera dimensión en el *EVD* para elaborar un planificador más completo.

En el método implementado no se permite seleccionar comandos que están dentro de los objetos de velocidad mapeados sobre el *EVD* porque conducen a colisión. Sin embargo, esta colisión no es inmediata ya que el mapeo se ha hecho dentro de un horizonte temporal. Se podría permitir ejecutar estos comandos de velocidad durante un tiempo determinado (unas cuantas iteraciones) para, por ejemplo, proporcionar movimientos más rápidos o permitir al robot posponer el reducir la velocidad en la situación *Frenar*,

obteniendo al final trayectorias más rápidas hacia el objetivo pero sin descuidar la seguridad.

- Incorporación de sistemas de percepción que proporcionen información más completa del entorno pero que su tratamiento no lleve más tiempo del periodo disponible.

Apéndice A

Herramientas utilizadas

En este apéndice se muestra el conjunto de herramientas utilizadas para el desarrollo de las distintas tareas llevadas a cabo durante la realización de este PFC.

- Desarrollo y pruebas

- Eclipse 3.5: Entorno de desarrollo.
- CGAL: Librería gráfica utilizada para definir los *ODVs* en el *EVD*. Proporciona algoritmos para poder hacer cálculos sobre ellos ([CGAL]).
- DISLIN: Librería gráfica empleada en el sistema para visualizar el *EVD* ([DISLIN]).
- Player/Stage: Sistema de simulación para sistemas multi-robot ([Player/Stage]). Player es un servidor que proporciona un interfaz para controlar los dispositivos del robot. El modelo cliente/servidor de Player permite que los programas de control de los robots puedan estar escritos en cualquier lenguaje de programación que soporte sockets TCP y que se ejecutarán como clientes en cualquier máquina con conexión de red con el robot.
Stage es un simulador multi-robot que se usa junto con Player. Proporciona un entorno en dos dimensiones para robots móviles y una variedad de modelos de sensores.
- Robot Pioneer: Robot de la familia Pioneer de ActivMedia disponible en el Departamento de Informática e Ingeniería de Sistemas. Se puede obtener más información consultando la página <http://www.mobilerobots.com>.

- Documentación

- Kile 2.0.3: Editor de \LaTeX .
- GNUPLOT 4.4: Herramienta gráfica para Linux.
- Gantt Project 2.0.10: Editor de diagramas de Gantt.
- Inkscape 0.48: Editor de gráficos vectoriales SVG de código abierto.

Apéndice B

Algunos conceptos

B.1. Robots no holonómicos

Para poder interaccionar con el mundo, el robot utiliza unos dispositivos que producen ciertos efectos en el entorno bajo el control del robot, se denominan *efectores*. Para producir un efecto en el mundo físico, el efector debe estar provisto de un *actuador* que permite convertir comandos de software en movimiento físico. Los actuadores define un sólo tipo de movimiento o grado de libertad como por ejemplo, un motor eléctrico o los neumáticos.

En este caso de aplicación, se trata de modificar la ubicación del robot. Un robot tiene tres grados de libertad, dos de su posición $x - y$, y una correspondiente a su dirección. Sin embargo, sólo son dos sus actuadores, la conducción del vehículo y el mecanismo de dirección. Es decir, podemos desplazarlo en la dirección a la que mira o girarlo, pero nunca desplazarlo lateralmente. Como la cantidad de grados de libertad controlables es de sólo dos, que es menor que la cantidad total de grados de libertad (tres), se habla de un robot no holonómico ([Robotica]).

B.2. Transformaciones

Dado que el robot se mueve, se debe considerar el movimiento en el seguimiento de los objetos. Cuando se aplica el *FKE* todas las variables deben expresarse en el mismo sistema de referencia. Por tanto, hay que trasladar los valores de las variables en el instante anterior al instante actual. Para eso se utilizan las matrices de transformación homogénea, que relacionan las variables del estado actual con las variables del estado anterior. Para más información, consultar [[Robotica]].

Apéndice C

Tratamiento de la información del sensor

En el apartado C.1 de este apéndice se hace una revisión al *Filtro de Kalman Extendido*. El proceso de segmentación de la información estática se explica en el apartado C.2.

La información que se dispone del entorno es esencial a la hora de planificar la navegación de un robot, ya que a partir de ella se construye el modelo sobre el cual se tomarán las decisiones de movimiento que darán lugar a las diferentes estrategias de navegación.

Para el desarrollo de un sistema de navegación se deben tener en cuenta otras cuestiones que simplemente la generación del movimiento. El problema de la localización del robot así como la detección y seguimiento de los objetos móviles son algunas de ellas. Afortunadamente, el grupo de Robótica disponía del software para su tratamiento, aunque fue necesario dedicar un tiempo a entender su funcionamiento, en especial el *Filtro de Kalman Extendido*, método empleado para la identificación y seguimiento de los obstáculos móviles.

C.1. Filtro de Kalman Extendido

El *Filtro de Kalman Extendido*, *FKE*, proporciona un buen marco para la estimación de una variable de la que se disponen medidas a lo largo del tiempo. El objetivo del *FKE* es obtener un estimador de las variables de estado de un sistema dinámico, basado en observaciones ruidosas y en un modelo dinámico de la evolución del sistema ([Welch y Bishop], [?]).

El *FKE* predice el estado de un obstáculo (posición y velocidad). A partir de la medida obtenida del láser corrige dicha estimación, dando mayor peso a la medida observada o la estimación realizada. Esto mismo se hace para la estimación y corrección de la covarianza del error.

Para este trabajo, se ha considerado que los obstáculos llevan velocidad constante. Sin embargo, ningún objeto real puede iniciar el movimiento con aceleración nula y terminarlo

sin sufrir cambios en la aceleración. Por eso, se considera el movimiento con velocidad constante y ruido blanco, que representa la aceleración que varía con el tiempo. El *FKE* permite modelar la aceleración como ruido independiente y blanco, es decir, el error en un momento dado no debería tener influencia sobre una medida posterior.

Para aplicar el *FKE* y conseguir un seguimiento fiable de un obstáculo, se debe proporcionar una característica que identifique al obstáculo en las sucesivas observaciones proporcionadas por el láser. A esta característica se le denomina *centroide* y se calcula como en [Santos y otros]. Se trata de un punto cuya coordenada en x es la del mínimo punto (punto de menor distancia al láser) y coordenada en y es la del punto medio de la nube de puntos que representa el obstáculo. Este punto será la medida a incorporar al filtro. Si no se asocia *centroide* a un obstáculo durante algunas iteraciones, deja de considerarse dicho obstáculo.

El método estima el estado x_{k+1} de un sistema que responde a la ecuación:

$$x_k = f(x_{k-1}, u_{k-1}) \quad (\text{C.1})$$

con una medida z_k correspondiente a la observación, que es:

$$z_k = h(x_k) + v_k \quad (\text{C.2})$$

El *FKE* aproxima las ecuaciones C.1 y C.2 utilizando el desarrollo en serie de Taylor para obtener las ecuaciones que finalmente se utilizan para estimar x_{k+1} .

Las ecuaciones del filtro pueden clasificarse en dos etapas, *predicción* y *actualización*. En la primera etapa se predice el siguiente estado del sistema x_k^- y la covarianza del error:

$$x_k^- = A_k x_k \quad (\text{C.3})$$

$$P_k^- = A_k \quad (\text{C.4})$$

donde la matriz A relaciona el estado entre el instante k y $k + 1$.

Las ecuaciones de actualización se encargan de incorporar la medida al estado estimado x_k^- (ecuación C.5). Se calcula primero la matriz de *ganancia de Kalman* (ecuación C.6), que establece la cantidad de influencia del error entre la estimación y la medida:

$$x_k = x_k^- + K(z_k - H_k x_k^-) \quad (\text{C.5})$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (\text{C.6})$$

con R_k la covarianza del error medido. Si R_k se aproxima a 0, quiere decir que podemos confiar en la medida, por lo que la ponderación del residuo será mayor. Por el contrario, si P_k^- se acerca a 0, se tendrá mayor confianza en la estimación ponderando menos el residuo.

La diferencia $(z_k - H_k x_k^-)$ se conoce como innovación de la medida o residuo y refleja la diferencia o discrepancia entre la predicción de la medida y la observación actual.

Finalmente, se actualiza la matriz de covarianza del error en el proceso:

$$P_k = (I - K_k H_k) P_k^- \quad (\text{C.7})$$

La inicialización de las matrices de covarianza del error en la medida R y del error del proceso Q suele hacerse por experimentación. Para el primero, analizando la traza de datos obtenidos off-line y para el segundo, un modelo sencillo (pobre) suele ser suficiente. Si Q y R son constantes, tanto la covarianza del error en la estimación P_k como la *ganancia de Kalman* K_k se estabilizarán rápidamente ([Welch y Bishop]).

C.2. Segmentación de la información estática

La información estática que se dispone del entorno consiste en un conjunto de puntos proporcionados por el láser. La técnica de segmentación implementada toma esos puntos e identifica los diferentes segmentos que compondrán los objetos estáticos.

El proceso seguido en el algoritmo es el empleado en [Castro y otros] y se explica a continuación. Se basa en el cálculo de la distancia entre dos puntos consecutivos, $d(P_i, P_{i+1})$:

$$d(P_i, P_{i+1}) = \|P_{i+1} - P_i\| = \sqrt{r_{i+1}^2 + r_i^2 - 2r_{i+1}r_i \cos \Delta\alpha} \quad (\text{C.8})$$

Si,

$$d(P_i, P_{i+1}) \leq C_0 + C_1 \min\{r_i, r_{i+1}\} \quad (\text{C.9})$$

con,

$$C_1 = \sqrt{2(1 - \cos \Delta\alpha)} \quad (\text{C.10})$$

el punto P_{i+1} pertenece al mismo segmento que el punto P_i . La constante C_1 es la distancia entre puntos consecutivos en cada observación. La constante C_0 permite ajustar el algoritmo de segmentación a ruido. Con estos dos valores se consigue captar puntos no seleccionados, como los de las esquinas.

Además, se determina que el número mínimo de puntos para formar un segmento sea 4, eliminando puntos aislados. Una vez que se tienen ya los segmentos, se alargan y engordan con el radio del robot para tener mayor margen de seguridad.

Para realizar un buen ajuste de los parámetros fue necesario probar el algoritmo primero en simulación y posteriormente en real. Los segmentos resultantes serán los utilizados en el proceso de modelado de los objetos estáticos sobre el *EVD*, apartado 2.3.

Apéndice D

Modelado del entorno

D.1. Fusión de objetos móviles

Inicialmente, y dado que se estaba utilizando la librería [CGAL] para representar las superficies de los *ODV* en la implementación del proceso de toma de decisión de estrategias, se pensó en emplearla y aprovechar las operaciones sobre superficies que proporcionaba para determinar el(los) *ODV(s)* resultante(s) de la unión de los diferentes *ODV*. Sin embargo, esta operación de unión era costosa y se optó por buscar una solución basada en el cálculo del modelo desarrollado, es decir, en la obtención de aquellas trayectorias con peligro de colisión y sus comandos de velocidad y tiempos asociados.

La primera aproximación consistía en calcular el *ODV* de cada obstáculo a partir de un mismo conjunto de radios. Para eso, se analizó cada intervalo de radios para el que se hacen los cálculos, y que vienen definidos a partir de los radios tangentes a las rectas que constituyen la *banda de colisión* de los obstáculos en el *EC*. Entonces, se escogía el menor de los positivos (giros a izquierda) y el mayor de los negativos (giros a derecha). La fusión implicaba únicamente la comparación de los diferentes comandos para un mismo radio y la selección de aquel con mayor velocidad lineal superior y menor velocidad lineal inferior. Sin embargo, se perdía resolución en los *ODV* obtenidos ya que el número de radios seleccionado para los intervalos era insuficiente para representar el dinamismo de los obstáculos, llevando a un *EVD* que no reflejaba todas las características identificadas hasta el momento, como el caso de las zonas de *Valle de Velocidad*. Al tener un mismo número de radios para un rango más ancho, se pierde potencia sobre el modelo.

Para subsanar este problema, se decidió aumentar el número de radios por intervalo, pero el sistema empleaba mucho tiempo en realizar los cálculos.

Finalmente, se decidió mantener el cálculo de intervalos de radios a colisión como hasta entonces y determinar si las trayectorias peligrosas con respecto a un objeto móvil lo eran también para el resto. El proceso es como sigue. Para cada obstáculo percibido se determina un conjunto de trayectorias que el robot podría describir, calculando las velocidades y los

tiempos a colisión asociados para construir el *ODV* correspondiente. Una vez se tiene esa información, se vuelven a realizar los cálculos para el resto de obstáculos considerados. El contorno superior del *ODV* resultante vendrá definido por el mayor de todos los comandos superiores obtenidos, y el contorno inferior, por el de menor velocidad. La figura 2.2 del capítulo 2 muestra la fusión de dos obstáculos.

D.2. Objetos estáticos

Como se menciona en el capítulo 2, se realiza un proceso de segmentación sobre los puntos estáticos proporcionados por el láser, dando lugar a segmentos de recta. Para proporcionar un margen de seguridad mayor, se decidió alargar y engordar dichos segmentos con el radio del robot.

Además, con el objetivo de simplificar el mapeo de los objetos estáticos en el *EVD*, se genera un número reducido de puntos equidistantes entre los extremos del segmento para aproximar la forma de los objetos en el *EVD* con un conjunto reducido de trayectorias circulares. Así, se disminuye la carga computacional del sistema.

La técnica empleada para modelar los objetos estáticos se basa en el concepto de distancia de frenado. Aplicado a este problema, supone una aproximación, ya que el robot se mueve describiendo clotoides y anti-clotoides, pero representa el peor caso, cuando describe una línea recta. Por lo tanto, $D_{seguridad}$ se define como la distancia necesaria para que el robot pueda frenar sin colisionar, navegando a máxima velocidad lineal. La ecuación es:

$$D_{seguridad} = \frac{v_{max}^2}{2a_v} \quad (D.1)$$

donde v_{max} es la velocidad lineal máxima que el robot puede llevar y a_v es la aceleración lineal propia del robot.

La distancia del robot al objeto estático se define como:

$$D_{colision} = atan2f(p_y, p_x) * radio_i \quad (D.2)$$

donde p_x, p_y son las coordenadas de un punto del segmento y $radio_i$ es el radio del arco de circunferencia para llegar a p_x, p_y .

Si $D_{colision}$ es mayor que $D_{seguridad}$, no hay peligro de colisión con el objeto estático y las velocidades que describen la trayectoria aparecen como libres en el *EVD*. Si no, se calcula la velocidad máxima que el robot podría llevar para poder frenar:

$$V_{frenado} = \sqrt{2 * a_v * D_{colision}} \quad (D.3)$$

$$W_{frenado} = \frac{V_{frenado}}{radio_i} \quad (D.4)$$

Además, para imponer mayor seguridad se determinó que si el robot estaba a una distancia muy cercana del objeto estático (a menos de 1 metro) se limitaba todo el EVD para el radio en concreto, dando valor 0 a $V_{frenado}$ y $W_{frenado}$.

Apéndice E

Estrategias de navegación

Este apéndice está dedicado a presentar las *variables de decisión* que influyen en la selección de las diferentes estrategias, así como explicar con más detalle las estrategias de navegación aplicadas en cada situación descrita en el apartado 4.1.

E.1. Variables de decisión

Para detectar una situación es necesario analizar el *EVD* y reconocer las características más relevantes del mismo una vez que el entorno se ha mapeado sobre él. A estas características se les denomina *variables de decisión*. Tras ser evaluadas, permitirán al robot tomar decisiones mientras navega hacia el objetivo. Las *variables de decisión* pueden ser inherentes tanto a los obstáculos como al robot. Aparecen mapeadas en el *EVD*, que es el espacio utilizado para establecer la estrategia de movimiento, y conducirán el proceso de decisión jerárquico en el árbol de decisión. Estas variables son:

Variables en el *EVD*.

- Velocidades *libres* y *no libres*. Las zonas fuera del *ODV* son zonas de velocidades libres. En concreto, la zona de color magenta de la figura E.1b corresponde a velocidades *libres* dentro de los límites del *ODV*; se denomina *Valle de Velocidad*.
- Zonas de velocidades *superiores* e *inferiores*, que se refieren, respectivamente, a la zona donde se alcanzan las velocidades más altas y a la zona de velocidades de frenado.
- *Velocidad de seguridad* (representada en marrón en la figura E.1d, *SafeVelocity*). Se trata del mayor valor de velocidad lineal que el robot puede llevar para poder frenar y evitar una colisión.
- *Límites de velocidad máximo Derecho e Izquierdo*, correspondientes a las velocidades máximas alcanzables por el robot sin que haya colisión (líneas azul y rojo en la figura E.1d). Se refieren a los *Radios Límites Derecho e Izquierdo* en la figura E.1c, *Lower Boundary Radius* y *Upper Boundary Radius*.

- *Velocidades máximas angular y lineal.*
- La *Dirección de orientación* hacia el objetivo (*OrientacionObjetivo*) y la *Dirección al objetivo* (*DirObjetivo*) para el movimiento del robot. En la figura E.1b la *OrientacionObjetivo* (*SteeringDir*) se ve mapeada como una línea vertical roja en un lateral, lo que significa que el robot está desorientado con respecto al objetivo, permitiendo que el ángulo de orientación pueda reducirse rápidamente. Si *OrientacionObjetivo* está dentro del *EVD*, la dirección mapeada del objetivo *DirObjetivo* (representado por una línea negra en la figura E.1b) se utiliza como radio de curvatura a alcanzar desde el estado actual.
- *Velocidad de alineación*, define la velocidad que el robot debería tomar para alinearse en la dirección de movimiento del obstáculo. En ciertas circunstancias ésta será la única forma de evitar una colisión (*V2* en la figura E.1f).
- *Velocidad actual* del robot, $(w, v)_R$, a partir del cuál se definirá la ventana dinámica del robot en el *EVD*.

Variables en el *ET*.

- La *posición relativa* entre el robot y los objetos, reflejando si el robot se encuentra antes, dentro o después de la banda de colisión.
- La *distancia angular* entre el robot y el objetivo.

E.2. Estrategias de navegación

E.2.1. MovimientoLibre

Se trata de la situación más segura en la que puede encontrarse el robot ya que no hay peligro de colisión con ningún obstáculo. El *EVD* no contiene mapeado ningún objeto móvil, por lo que se aplica directamente un criterio de alineación con el *goal*.

La estrategia de movimiento consta de varias partes. La primera consiste en un movimiento de aceleración angular máxima, tramo *V1* – *V2* en la figura E.2b, seguido por un movimiento de deceleración angular máxima, tramo *V2* – *V3* en la figura E.2d, describiendo curvas de tipo clotoide para orientarse en la dirección del *goal*. Se trata de controles extremos en espacio libre para alinear el robot y el *goal* ([Renaud y Fourquet] y [Balkcom y Mason]). Cuando el robot está orientado en la dirección del *goal*, es decir, cuando *OrientacionObjetivo* está dentro del *EVD* y la dinámica del robot está situada sobre la línea negra en el *EVD*, se aplica entonces un comando de velocidad lineal a máxima aceleración, tramo *V3* – *V4* en la figura E.2f. Como resultado, la dinámica del robot ya no está sobre *DirObjetivo*, por lo que tiene que ir a buscarlo ejecutando un comando de aceleración angular máxima (figura E.2h). Finalmente, aplica velocidades a máxima aceleración lineal (tramo de línea recta en el *EVD* de la figura E.2j,

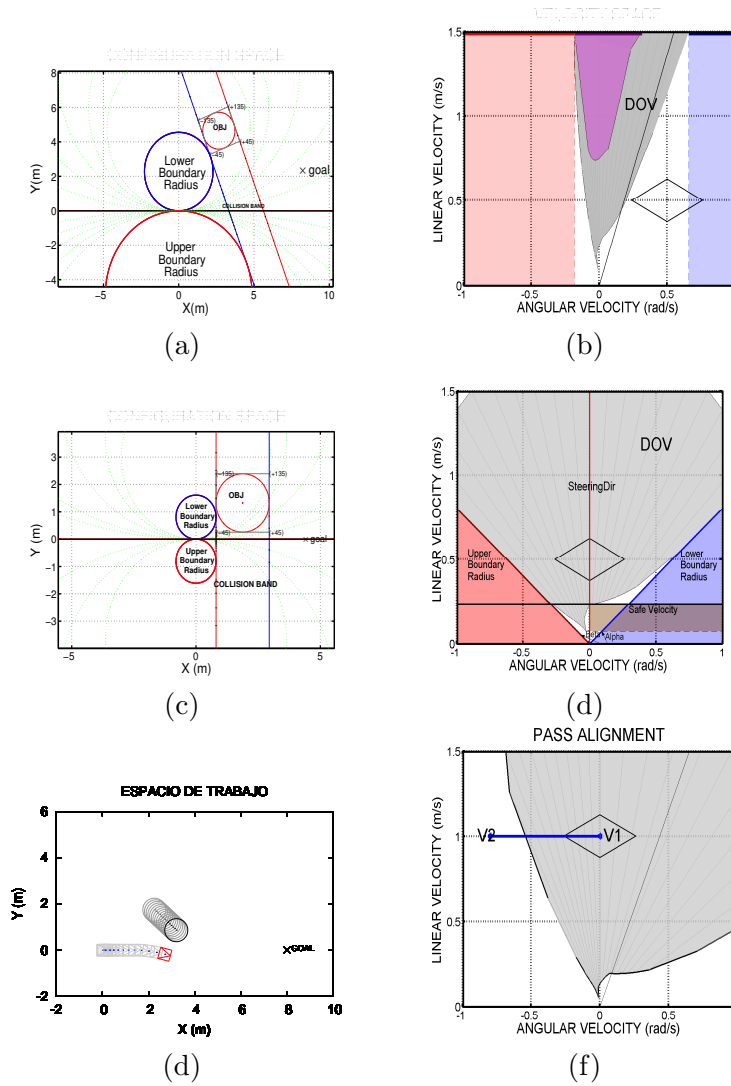


Figura E.1: (a) Espacio de configuración donde el robot podría pasar por delante del objeto móvil, (b) El mismo caso en el *EVD* muestra un *Valle de Velocidad* (velocidades superiores libres); el rombo define la ventana dinámica del robot y representa las aceleraciones angular y lineal máximas para un robot diferencial, cuyo centro es la velocidad actual. (c) En este caso el robot puede maniobrar permitiendo al obstáculo pasar antes que el robot en el *EC*, (d) La misma situación en el *EVD*. (e) Escenario donde el robot puede evitar la colisión ejecutando velocidades angulares en la dirección de movimiento del obstáculo, (f) situación en el *EVD*.

$V5 - V6$, correspondiente a una anti-clotoide), alcanzando velocidad máxima y reduciendo el tiempo en alcanzar el *goal*. La estrategia se reanuda en cada periodo de tiempo para mantener la alineación con el *goal*.

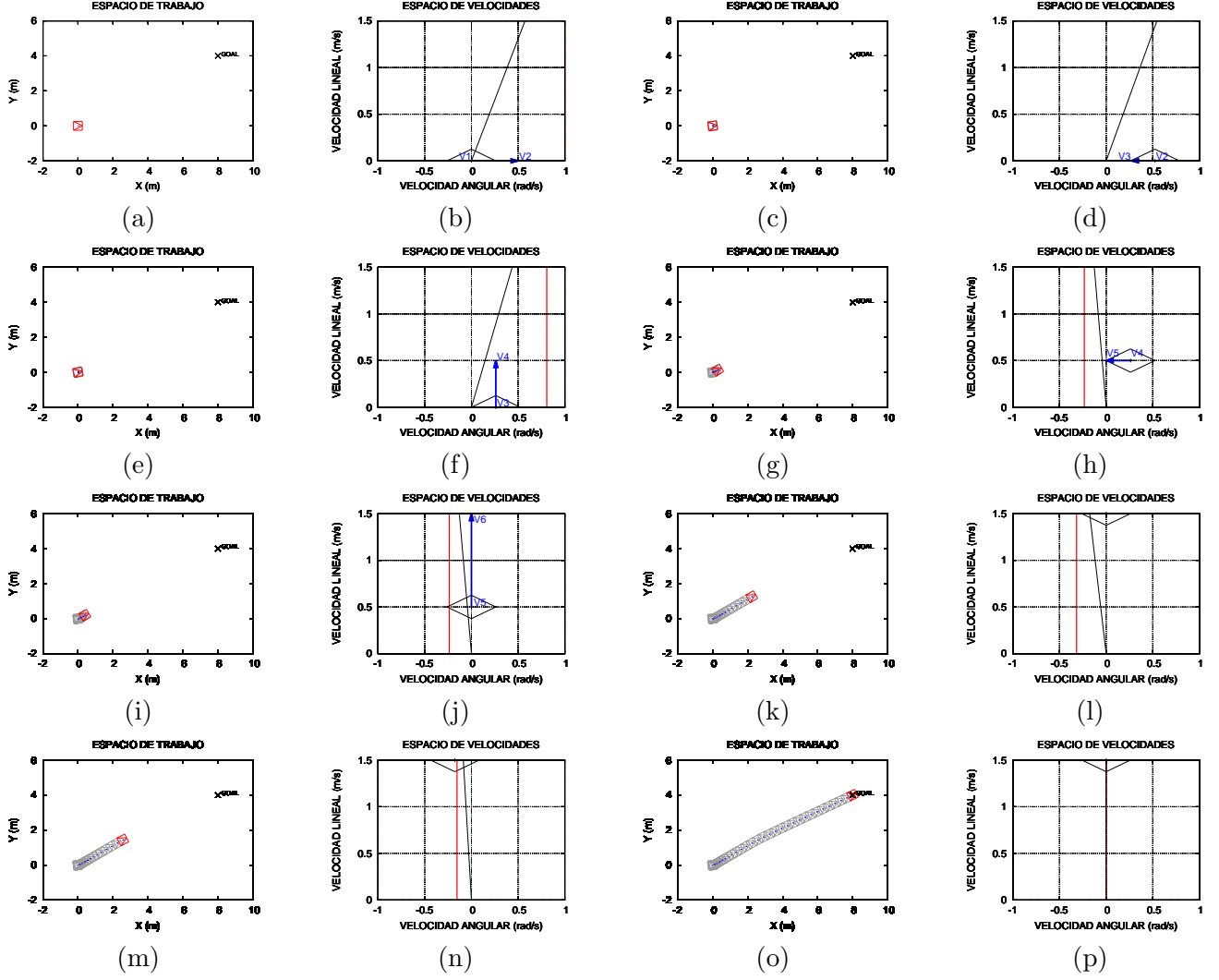


Figura E.2: Evolución de la situación *MovimientoLibre*. La línea roja vertical representa la *OrientacionObjetivo* que debe seguir el robot para alinearse con el *goal*, y la negra es el *DirObjetivo* a ser alcanzado una vez está orientado hacia el *goal*.

E.2.2. Pasar Por Delante

Se refiere al conjunto de estrategias que llevan al robot a pasar por delante del obstáculo.

Como ya se menciona en la sección 4.1, el aplicar comandos de velocidad y aceleración máximos en espacio libre conduce a trayectorias de tiempo mínimo. Sin embargo, no siempre es posible en escenarios con obstáculos, estáticos o dinámicos. Dependiendo de la estrategia que se esté ejecutando, si un comando de aceleración máxima es *no libre*, podrá o no ejecutarse. Para determinar si un comando de velocidad es *libre* o *no libre* se comprueba si el comando

está dentro o fuera de algún *ODV* en el *EVD*. Por ejemplo, en la figura E.1b, la zona superior en color magenta dentro del *ODV* representa un conjunto de velocidades lineales máximas que se pueden alcanzar, pero no para todas las velocidades angulares, aunque para llegar a dicha zona puede que haya sido necesario ejecutar comandos *no libres*, tal y como se verá más adelante.

PasarPrimero

Representa una situación en la que un conjunto de velocidades en la zona más alta del *EVD* dentro del *ODV* aparece libre de colisión (*Valle de Velocidad*), lo que significa que el robot puede cruzar por delante del objeto móvil. La figura E.3 refleja esta situación.

En base a un criterio de seguridad, se calcula el comando de velocidad $V4$ que deberá seguir el robot para resolver la situación. $V4$ se mapea en la zona de velocidades *superiores libres* dependiendo tanto del valor de velocidad *inferior* del *Valle de Velocidad*, que define su profundidad dentro del *EVD*, como del *DirObjetivo*. Para alcanzar $V4$, el robot debe describir primero una trayectoria de tipo anti-clotoide, $V1 - V2$ (figura E.3b), para situarse en velocidades *libres* dentro del *Valle de Velocidad*, seguido de una clotoide, $V2 - V3$ (figura E.3d), para situarse sobre el *DirObjetivo* recalculado, y por otra anti-clotoide $V3 - V4$ (figura E.3f), llegando a velocidad lineal máxima. Una vez que se alcanza $V4$, el robot mantendrá este comando de velocidad hasta que se seleccione una estrategia apropiada para resolver una nueva situación (figura E.3h-n). Esto garantiza una curvatura continua durante toda la maniobra.

PasarAlineado

En esta situación, el *ODV* del obstáculo móvil deja una zona de velocidades libres de colisión en uno de los lados del *EVD*, correspondiente a aquellas velocidades que conducen al robot en la misma dirección de movimiento que la del objeto móvil. La seguridad es prioritaria, por lo que el robot debe aplicar un control extremo para escapar de esta situación alineándose con el objeto móvil para poder pasar por delante de él y alcanzar el *goal*.

En la figura E.4 se muestra la maniobra completa para resolver esta situación. El robot debe tomar velocidades angulares que lo lleven en la misma dirección de movimiento del obstáculo para evitar la colisión con él. En primer lugar, se calcula un comando de velocidad $V3$ en la zona de velocidades *libres* a seguir por el robot para escapar de las velocidades de colisión, llevándolo a aumentar su velocidad angular y a describir una trayectoria de tipo clotoide, $V1 - V2$ (figura E.4b). A continuación, realiza una anti-clotoide (tramo $V2 - V3$ de la figura E.4d) para llegar a $V3$. En el momento en el que el robot alcanza $V3$ (figura E.4f), el comando de velocidad se mantiene hasta que la situación cambia (figura E.4h-n). Finalmente, se alcanza el *goal* a máxima velocidad lineal.

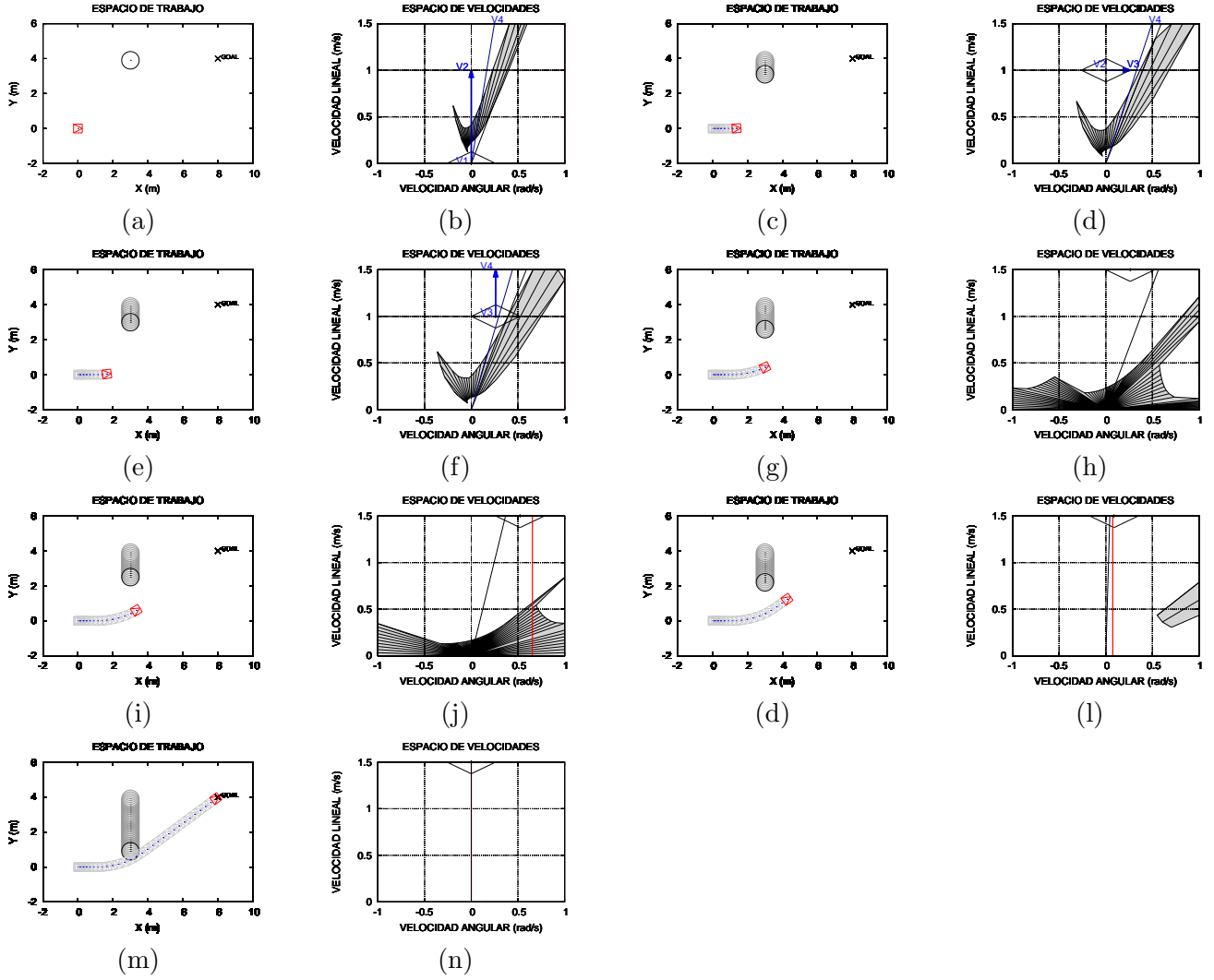


Figura E.3: Situación *PasarPrimero*. Las velocidades más altas están libres y pueden seleccionarse de manera que el robot pasa antes que el objeto móvil.

E.2.3. Pasar Por Detrás

Engloba el conjunto de estrategias que llevan al robot a pasar por detrás del obstáculo. Lo más importante a destacar es que cuando las velocidades lineales más altas en el *EVD* no son alcanzables (figura E.5), la única posibilidad para el robot es frenar para dejar que el obstáculo pase primero.

Frenar

Esta situación corresponde al caso en el que las velocidades altas en el *EVD* no pueden seleccionarse porque conducen a colisión. El *ODV* del objeto móvil mapeado ocupa casi todo

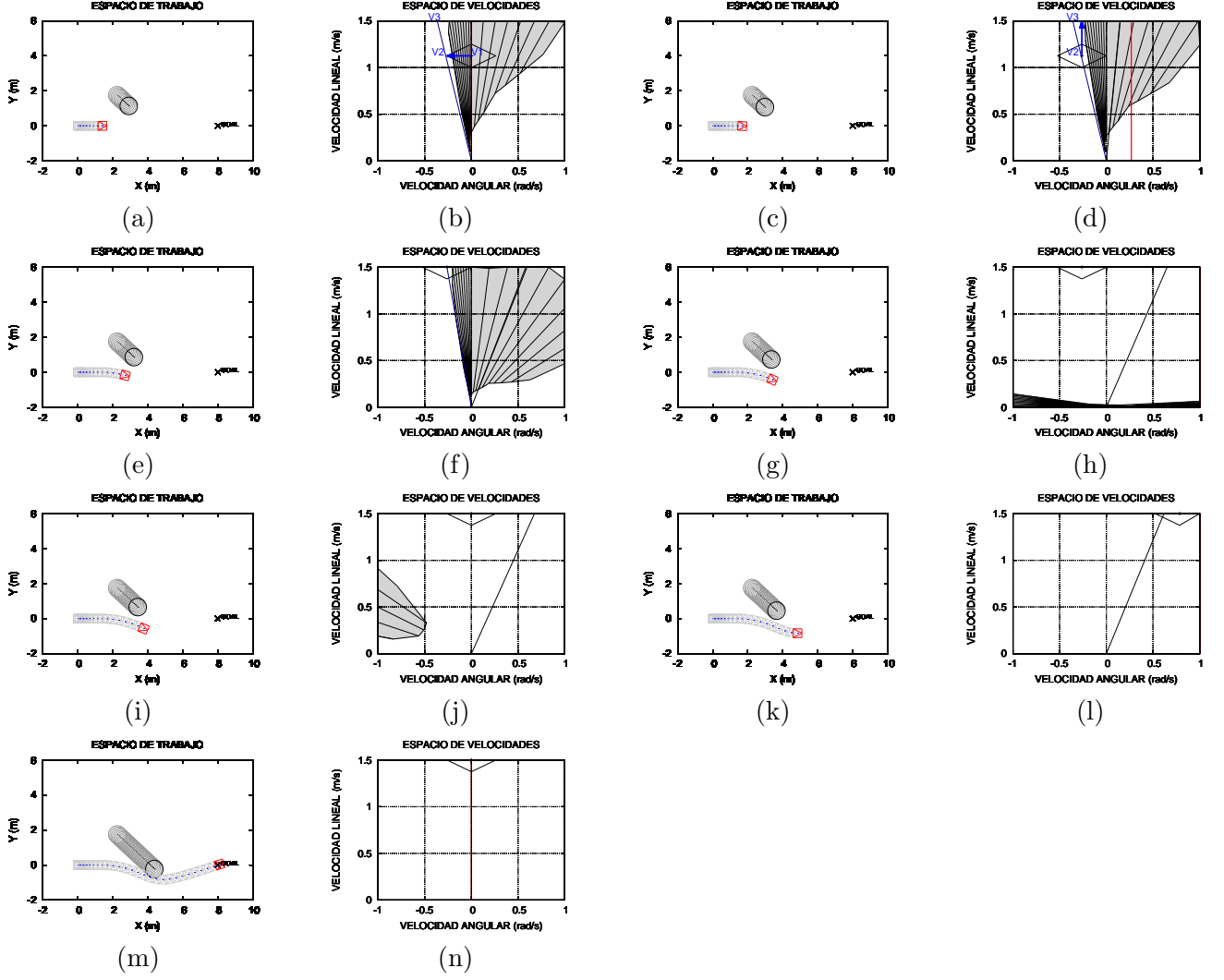


Figura E.4: Situación *PasarAlineado*. El robot tiene que alinearse en la dirección de movimiento del objeto para evitar la colisión pasando por delante del objeto.

el espacio de velocidades (figura E.5b), forzando al robot a buscar velocidades libres. Dado que las velocidades más altas son prohibidas, se aplica una solución que emplea un criterio de seguridad para evitar el objeto móvil. El robot debe maniobrar ejecutando velocidades lineales bajas para permitir al objeto pasar antes.

La estrategia seguida en el espacio de velocidad es la siguiente. En primer lugar, el robot reduce su velocidad lineal actual a máxima deceleración para escapar de las velocidades altas, las cuáles están ocupadas por el obstáculo en el *EVD*, hacia velocidades inferiores y seguras, desde V_1 a V_2 (figura E.5b). El movimiento que describe corresponde a una anti-clotoide en el *ET* y a una línea vertical en el *EVD*. A continuación, reduce su distancia angular

con el *goal* mediante una maniobra en dos pasos: la primera consiste en un movimiento con aceleración angular para alcanzar $V3$, que es una velocidad segura (figura E.5d); la segunda se lleva a cabo mediante un movimiento de deceleración angular para alcanzar $V4$ (figura E.5f). Ambos caminos son tramos de clotoides que permiten al robot alinearse con el *goal* a máxima aceleración angular. Después, el robot ejecuta comandos de máxima aceleración lineal conforme el obstáculo libera velocidades del *EVD*, describiendo tramos de anti-clotoide (figuras E.5h-l), hasta se produce un cambio de situación (figuras E.5n-r). Finalmente, se alcanza el *goal* a máxima velocidad lineal.

Evitando Objeto

Se trata de una situación en la que el objeto móvil se encuentra en el centro del *EVD*, dejando dos zonas de velocidades libres de colisión en la parte más alta, una a cada lado del *ODV*. La estrategia selecciona una de las zonas de velocidades libres en función de la dirección de movimiento del objeto móvil y de la dirección del *goal*, de manera que la maniobra resultante lleva al robot a pasar por detrás del objeto.

En la figura E.6 se muestra esta situación. $V4$ se mapea fuera de la zona ocupada por el objeto móvil en base a un criterio de proximidad al *goal* (variable *DirObjetivo*). La trayectoria final dependerá de la secuencia de movimientos ejecutados. Siempre que sea posible se ejecutarán comandos de máxima aceleración lineal, tramo $V1 - V2$ de la figura E.6b. Si el comando es *no libre* se realizará un movimiento de máxima aceleración o deceleración angular en dirección al *goal* para no desalinearse ($V2 - V3$ en la figura E.6d). Conforme el robot aumenta su velocidad lineal (tramo $V3 - V4$ en E.6f) el *ODV* evoluciona pasando a ocupar velocidades en uno de los lados del *EVD* (figura E.6h). Finalmente, se alcanza el *goal* a máxima velocidad lineal (figuras E.6j-l).

Ante un *EVD* como el de la figura E.6b, hay que tener en cuenta que, dependiendo de la situación relativa actual entre el *goal* y el robot, puede que el movimiento del robot le lleve a cambiar a una situación *PasarAlineado*. Este caso se daría cuando el obstáculo se moviera en la dirección donde está el *goal*, y el robot, al ejecutar movimientos hacia el *goal* estaría también ejecutando comandos de alineación con el obstáculo.

E.2.4. Dentro de la Banda de Colisión

En este tipo de situaciones el robot se encuentra dentro de la *BC* del obstáculo, por lo que debe salir cuanto antes para escapar de la colisión, aunque no siempre será posible. El *EVD* para este tipo de situación puede venir representado de varias maneras según la posición y velocidad relativas del robot y del obstáculo.

Si el *EVD* está totalmente ocupado por un *ODV*, se produce una situación *ColisionSegura* y el robot ya no tiene tiempo de reaccionar para evitar la colisión. La figura E.9 refleja esta situación.

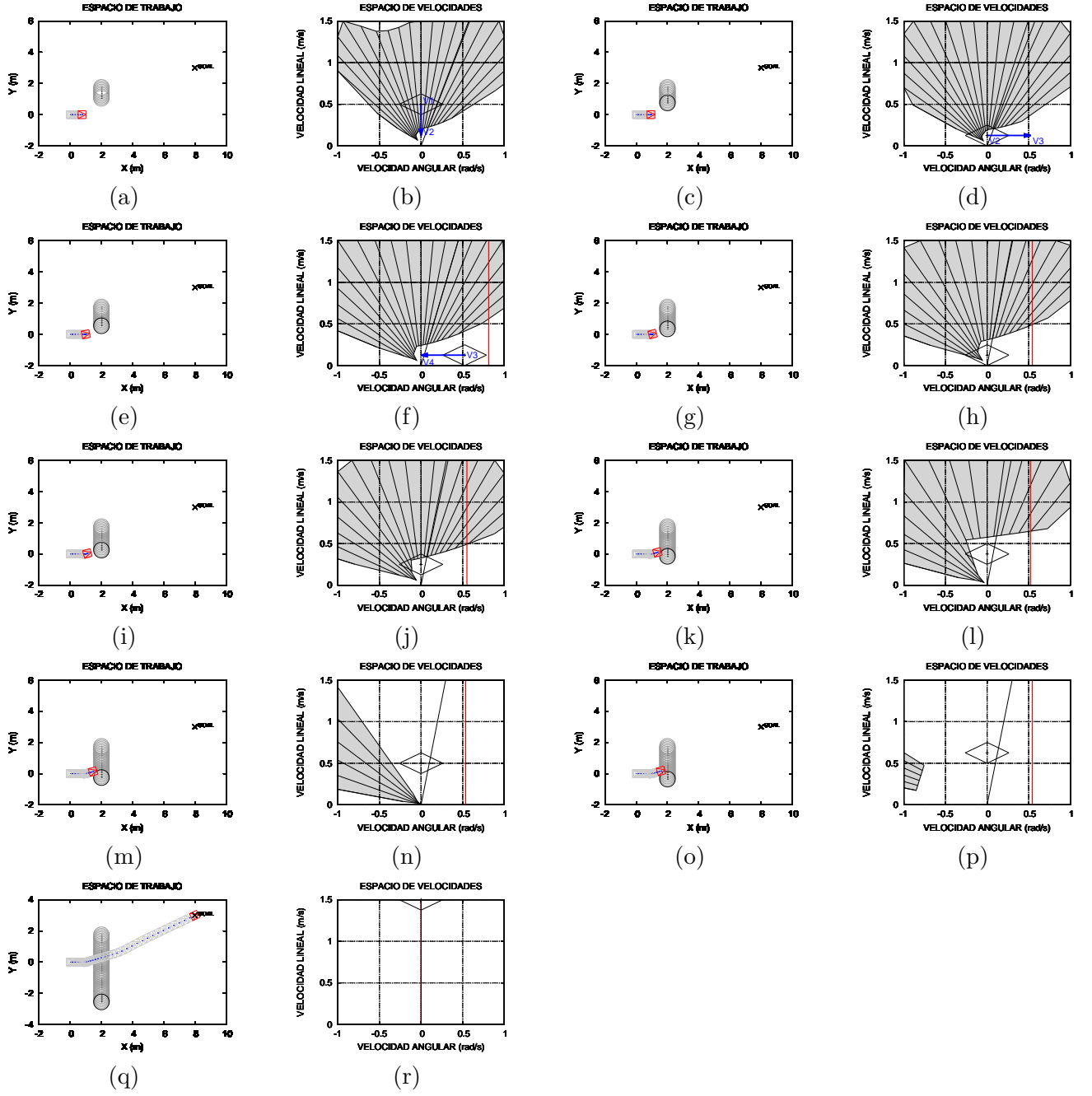


Figura E.5: Situación *Frenar*. El robot tiene que frenar para evitar la colisión y alcanzar velocidades seguras, permitiendo al obstáculo pasar primero.

Sin embargo, cuando el *EVD* aparece parcialmente ocupado por algún *ODV*, el robot puede resolver la situación.

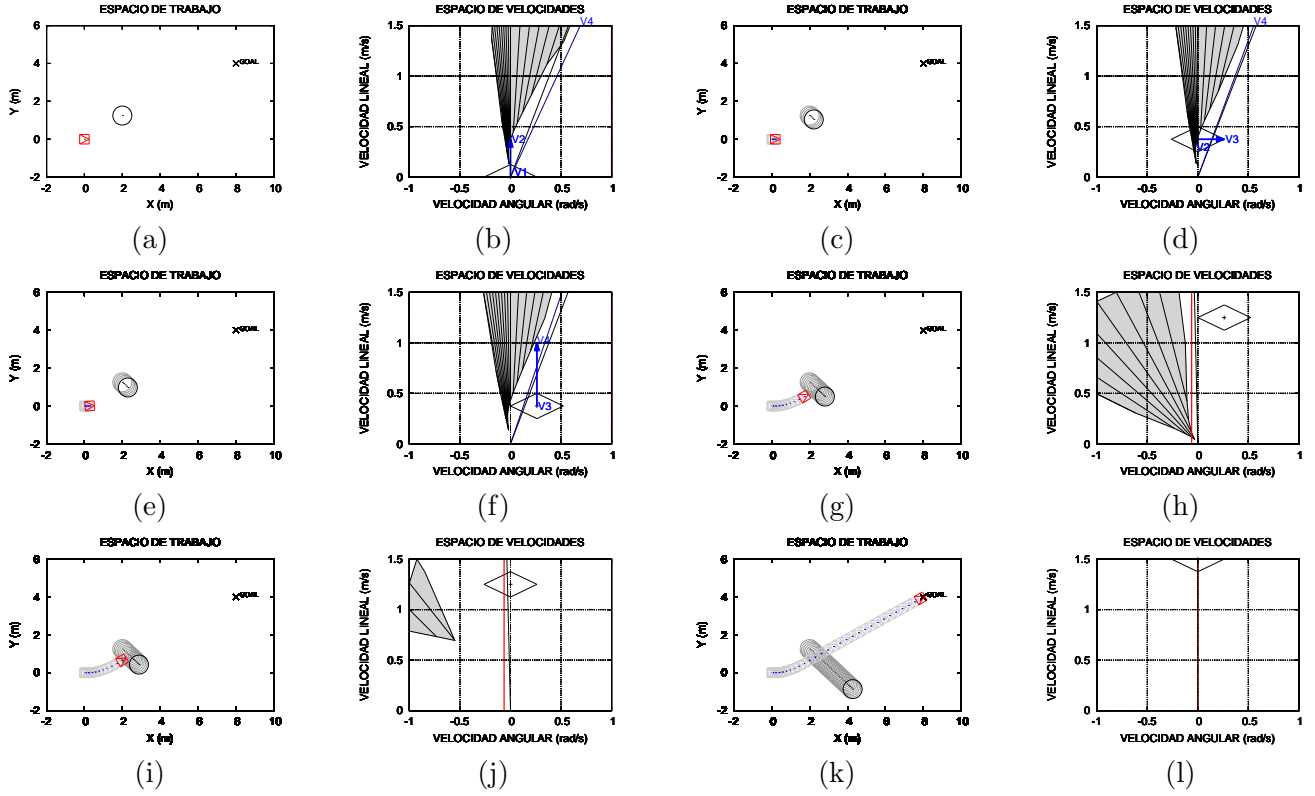


Figura E.6: Situación *EvitandoObjeto*. El robot maniobra pasando por detrás del objeto.

En la figura E.7d se muestra un *EVD* en el que todas las *velocidades inferiores* y un conjunto de las *velocidades superiores* aparecen como *no libres*. En este caso, el robot puede evitar la colisión aplicando comandos de máxima aceleración lineal para salir de la *BC*, a la vez que intenta no desorientarse de la dirección del *goal*. La situación que se plantea para este caso es la de *MovimientoLibre* ya que se emplea el mismo criterio para la selección de movimientos, la alineación con el *goal*, aunque la estrategia requiere de lógica adicional para recalculer el *DirObjetivo* en zona no ocupada y comprobar que el comando calculado se puede ejecutar. En la figura E.7d el robot describiría una anti-clotoide para alcanzar *V2*, ejecutando comandos de máxima aceleración lineal y escapando de la zona de peligro. Entonces, aplica un movimiento de máxima aceleración angular para situarse sobre *DirObjetivo* (tramo de clotoide *V2 – V3* de la figura E.7e) y alcanzar el *goal* en el menor tiempo posible.

La figura E.8 muestra otra situación con un *EVD* cuyas *velocidades superiores* son libres, mientras que las velocidades más bajas están ocupadas por el *ODV* del obstáculo. El robot actuaría de la misma manera que en el caso anterior, aumentaría su velocidad lineal a máxima aceleración para salir de la zona ocupada cuanto antes y dirigirse hacia el *goal*. La situación es también *MovimientoLibre*.

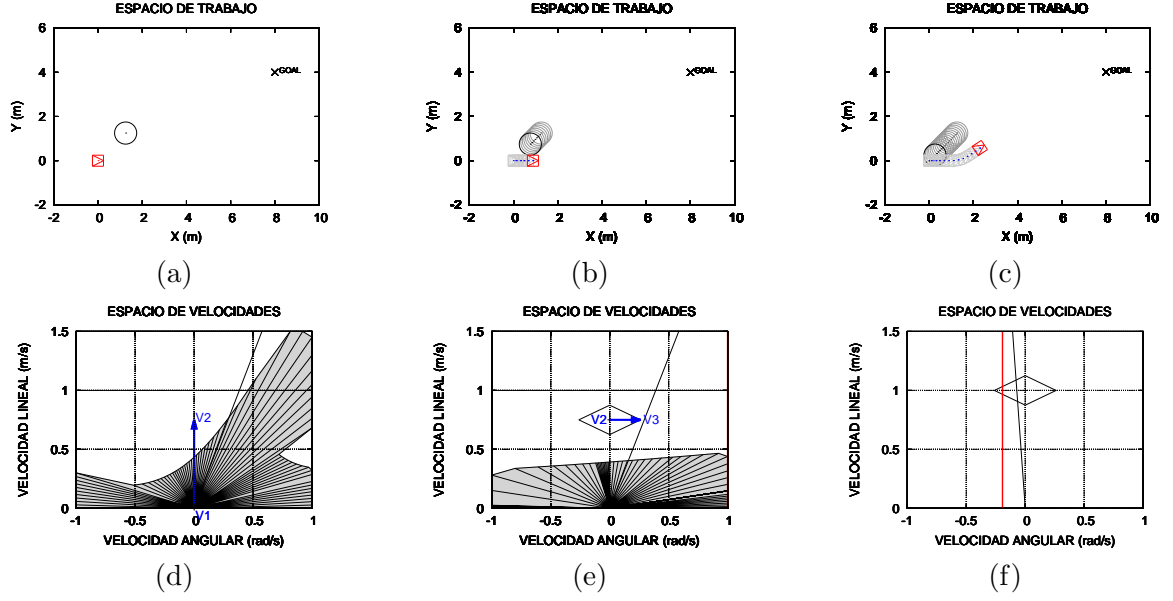


Figura E.7: Dentro de la banda de colisión. El robot escapa de la BC del obstáculo ejecutando de máxima aceleración lineal.

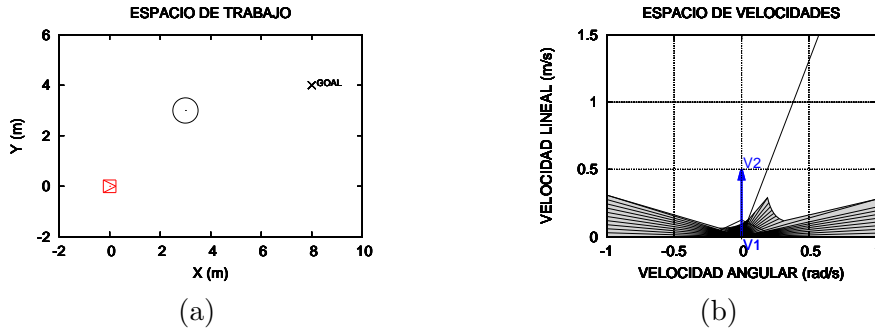


Figura E.8: Dentro de la banda de colisión. La zona de velocidades más alta está completamente libre, el robot escapa de la BC a máxima velocidad.

E.2.5. Varios obstáculos

En la figura E.10c se muestra un *EVD* con un *Valle de Velocidad* como resultado de la fusión de dos *ODVs*. La estrategia seguida por el robot consiste en un movimiento de tipo anti-clotoide aumentando su velocidad lineal para alcanzar el valle ($V1 - V2$ en E.10b,c). En ese momento, el robot está ya en una zona segura y sigue un criterio de alineación con el *goal*, recalculado dentro del *Valle de Velocidad* ($V2 - V3$ en la figura E.10e,f y $V3 - V4$ en E.10h,i) hasta que pasa por delante de los dos obstáculos y puede continuar sin peligro hacia el *goal*.

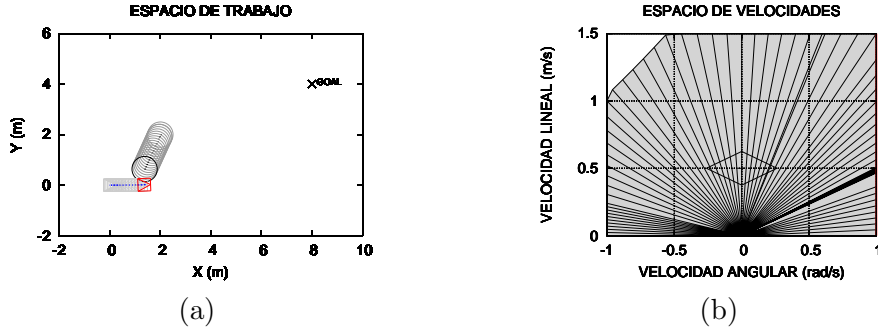


Figura E.9: Situación *ColisionSegura*. El robot no puede escapar a la colisión.

El escenario de la figura E.11a ilustra dos objetos móviles. El *EVD* resultante queda reflejado en E.11b, donde se distinguen dos *ODVs*. Uno de ellos contiene un *Valle de Velocidad* mientras que el otro aparece ocupando casi todo el espacio, por lo que el planificador selecciona *Frenar* para mantener al robot en velocidades bajas y esperar a que se libere espacio del *EVD*. Mientras tanto, ejecuta comandos de máxima aceleración y deceleración angular para alinearse con el *goal* (figuras E.11e-i), describiendo tramos de trayectoria de tipo clotoide. Llega un momento en el que la situación cambia a *PasarPrimero* como consecuencia de la detección de un *Valle de Velocidad* en el *EVD* (figuras E.12b,c), conduciendo al robot a pasar por delante del obstáculo y alcanzar el *goal* a máxima velocidad lineal (figuras E.12e-k). En el *ET* de la figura E.12j se ve cómo el robot pasa por detrás del obstáculo más cercano y por delante del más lejano.

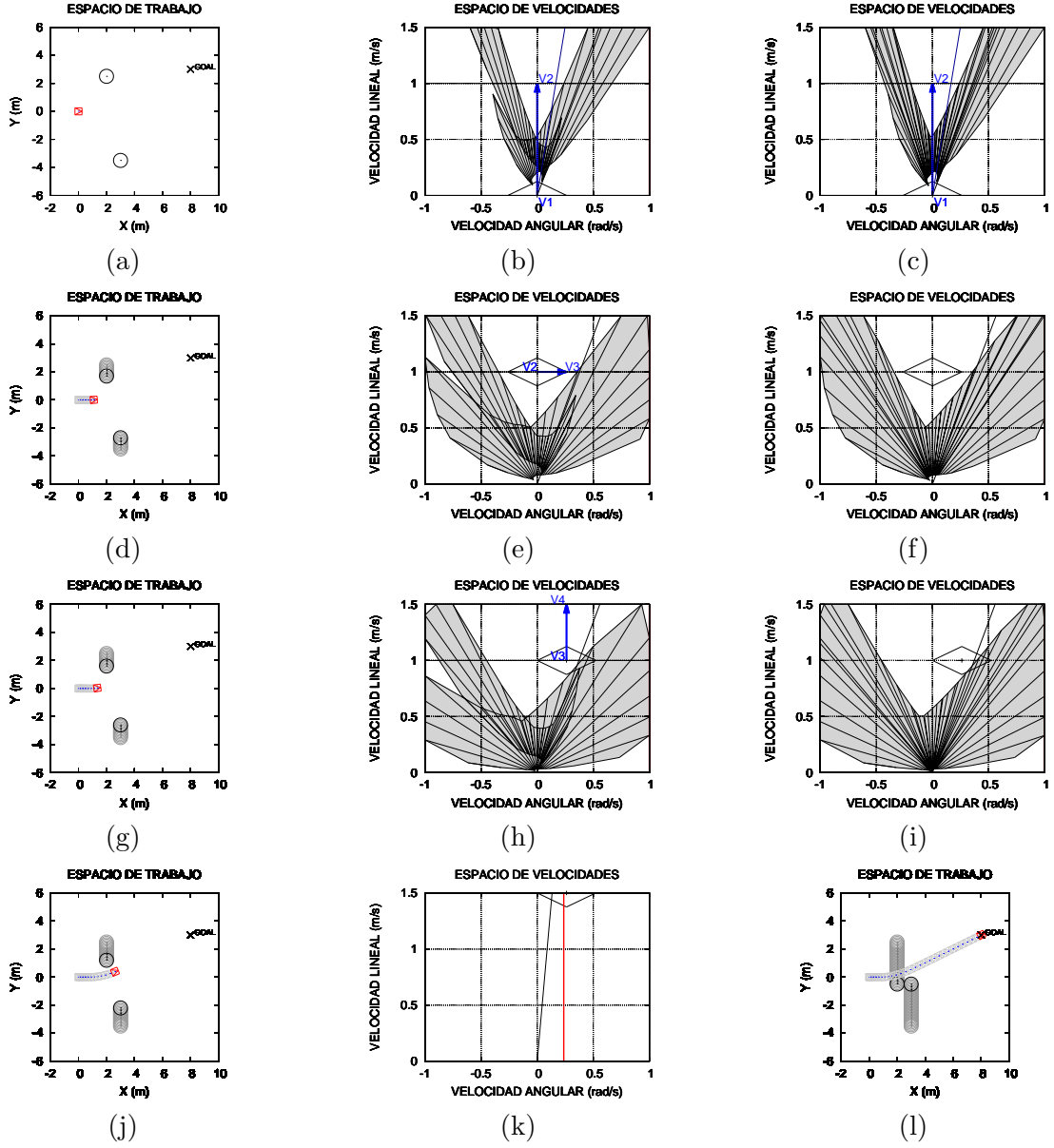


Figura E.10: Situación *PasarPrimero*. La columna del centro muestra el *EVD* con los *ODVs* sin fusionar, mientras que la de la derecha refleja el *ODV* resultante de la fusión. El robot pasa por delante de los dos obstáculos.

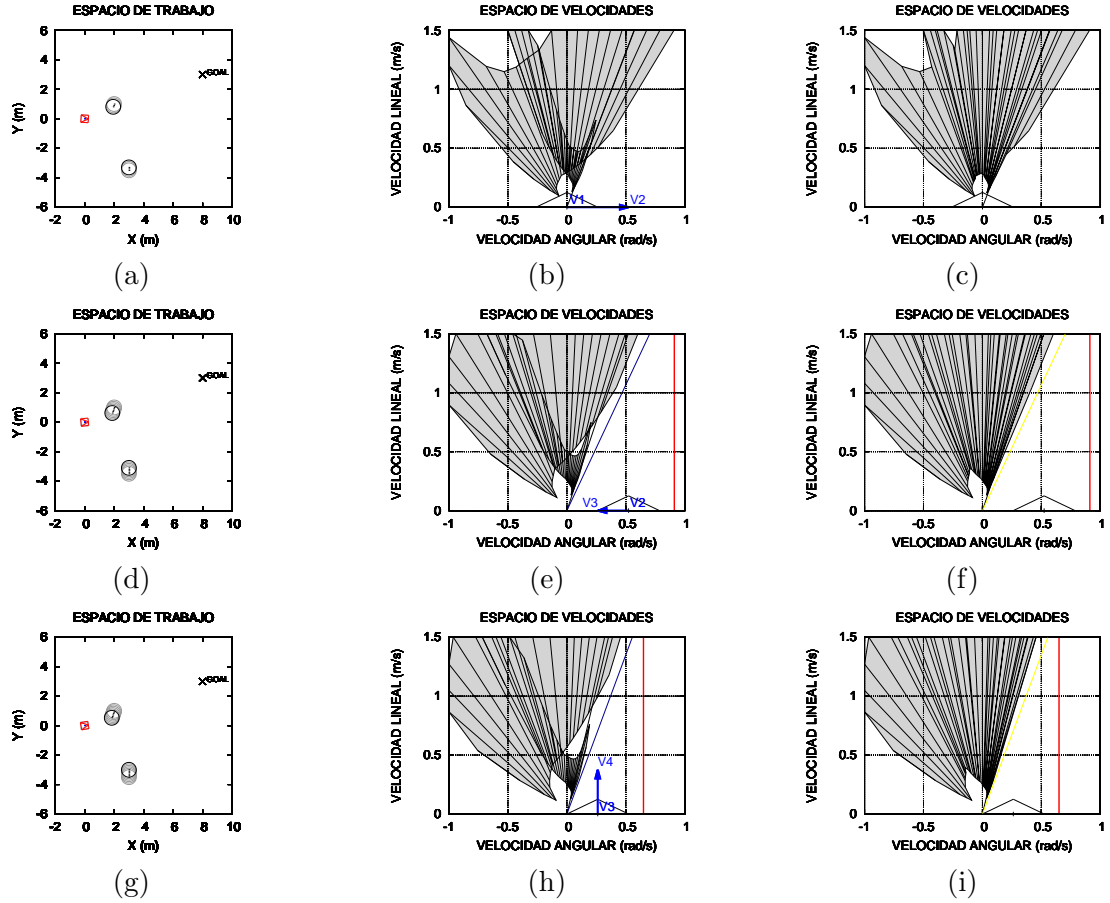


Figura E.11: Situación con varios obstáculos. Las columnas central y derecha muestran el *EVD* sin fusión y con fusión de los *ODVs* respectivamente. El robot se mantiene en velocidades bajas hasta que el *obj1* pasa, situación *Frenar*.

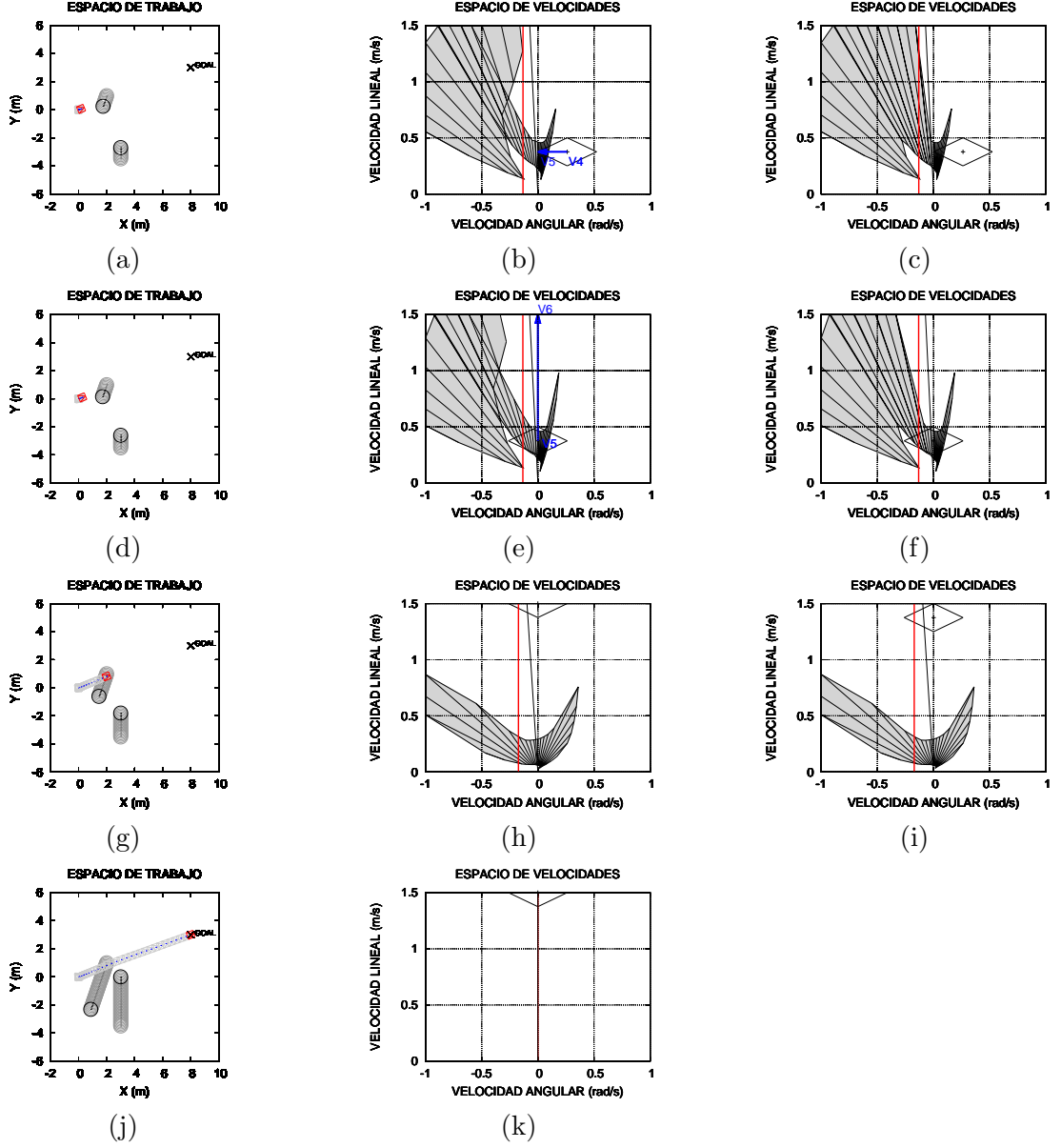


Figura E.12: Situación con varios obstáculos. Una vez que el primer obstáculo ha pasado, el robot ejecuta *PasarPrimero* para pasar por delante del segundo objeto.

Apéndice F

Simulaciones

En este anexo se amplia la información sobre los resultados obtenidos en las simulaciones.

F.1. Caos

La figuras F.1 y F.2 muestran imágenes sobre el *ET* y el *EVD* durante la navegación, sin fusión y con fusión de los *ODVs*.

La fusión de los objetos *obj2*, *obj4* y *obj5* (figura F.1c) da como resultado dos zonas tipo *Valle de Velocidad*. El valle que contiene velocidades libres de giros a izquierda (velocidades angulares positivas) es común a los obstáculos *obj2* y *obj4*, por lo que el tomar velocidades en esa zona llevaría al robot a pasar por delante de ambos. El valle de giros a derecha (velocidades angulares negativas) conduciría a pasar por delante del *obj2* y detrás del *obj4* ya que sólo pertenece al *obj2*.

Siguiendo un criterio de seguridad, el planificador selecciona el valle más profundo para realizar la maniobra y calcula el *DirObjetivo* en esa dirección (situación *PasarPrimero*). El robot aumenta su velocidad lineal para alcanzar velocidades dentro del valle (F.1e,f) aunque el *obj5* impide seleccionar comandos de velocidad angular negativa. En el momento en el que el *obj5* desaparece del *EVD* se libera un conjunto de velocidades que permiten al robot realizar un movimiento de giro hacia la derecha para situarse dentro del valle (figura F.1h,i).

A medida que el robot avanza, el *obj5* reaparece en el *EVD*, aunque no supone peligro para el robot porque está fuera de su *OVD* y no ocupa *velocidades superiores* como muestra la figura F.1k,l, siguiendo en *PasarPrimero*. Este movimiento hace que desaparezcan los objetos *obj2* y *obj5* y que se modele el *obj3*, figura F.2b,c. La situación pasa a ser *EvitandoObjeto*, aumentando la velocidad lineal del robot, hasta que en el *EVD* de la figura aparece un nuevo *Valle de Velocidad* correspondiente al *obj3* y se selecciona *PasarPrimero* (imágenes F.2e,f). Finalmente, se resuelve la navegación con el *obj4* ejecutando *EvitandoObjeto* y alcanzando el *goal* a máxima velocidad lineal (figuras F.2g-l).

Observamos que los obstáculos *obj1* y *obj6* no aparecen reflejados en el *EVD* durante la simulación ya que el planificador considera que no suponen peligro alguno para la navegación del robot.

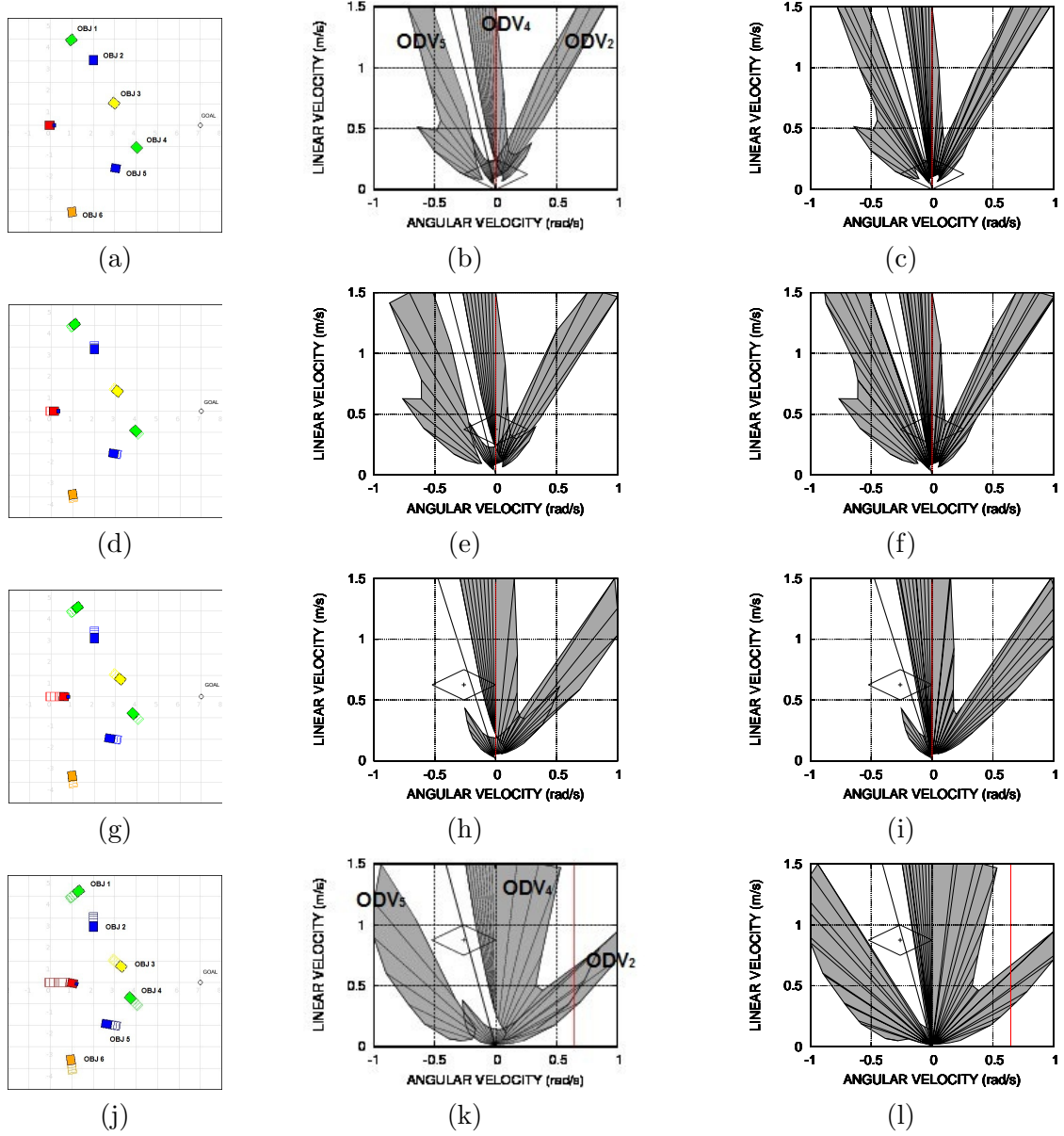


Figura F.1: Evolución del *ET* y del *EVD* durante la simulación *caos*.

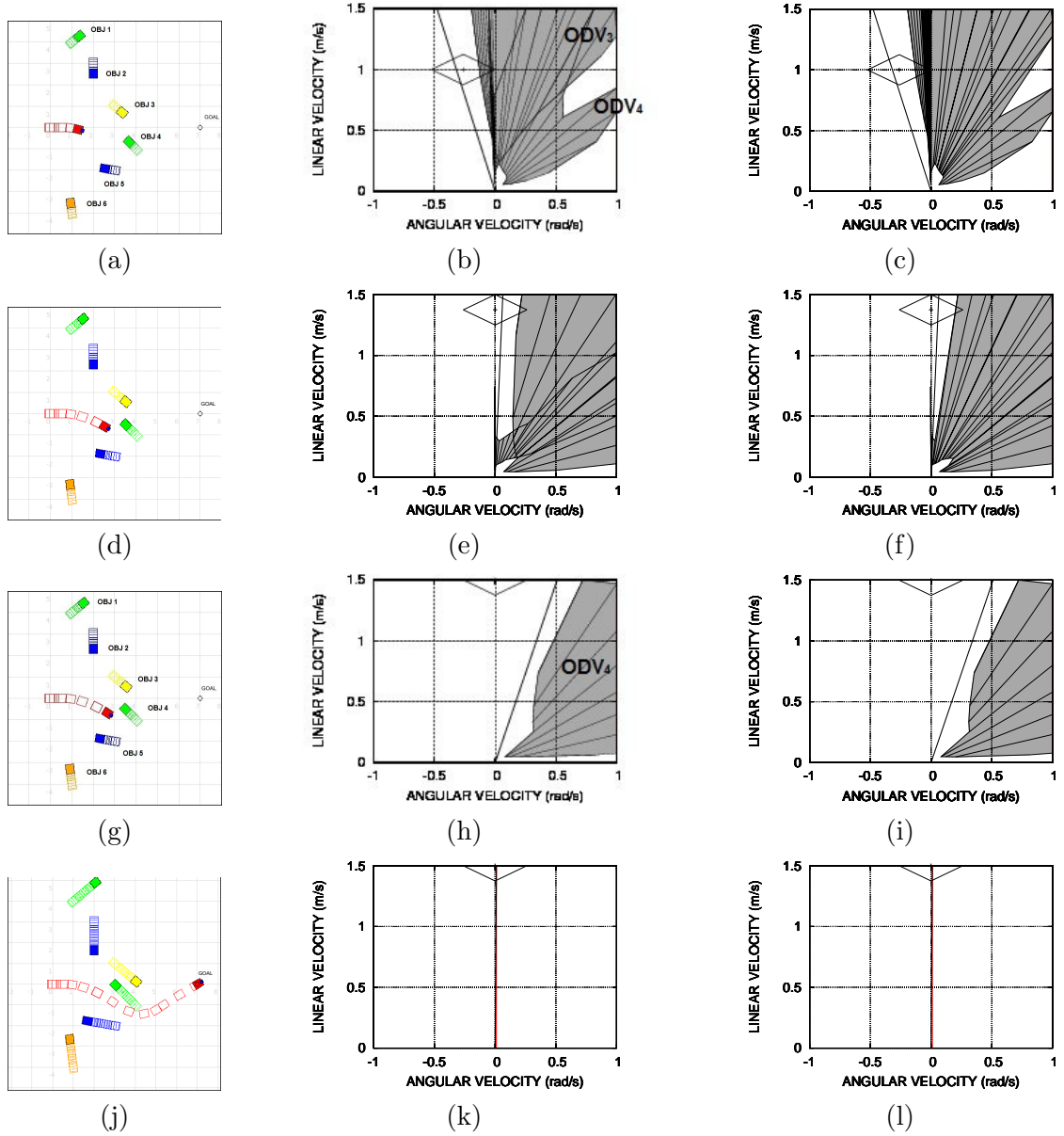


Figura F.2: Continuación de la evolución del ET y del EVD durante la simulación *caos*.

F.2. Carriles

Las figuras F.3 y F.4 muestran imágenes sobre el ET y el EVD durante la navegación, sin fusión y con fusión de los $ODVs$.

En la figura F.3a se representa la situación inicial a la que se enfrenta el robot. Dado que son muchos obstáculos, el planificador debe dar prioridad a los que suponen mayor peligro

y resolver primero la navegación con ellos. Por lo tanto, en el *EVD* de la figura F.3b,c aparecen mapeados sólo los obstáculos verde (*obj1*) y azul (*obj2*). Se observan dos zonas tipo *Valle de Velocidad*. El valle que aparece en el centro es común a los dos objetos dinámicos, mientras que el otro valle pertenece sólo al *obj1*. El planificador identifica la situación *PasarPrimero* y selecciona el valle más profundo para calcular la maniobra. Sin embargo, dado que los obstáculos llevan velocidades altas (0.5 m/s), los *ODVs* aumentan en seguida su tamaño (figura F.3e,f), lo que provoca que el robot haga un giro en la misma dirección de movimiento del obstáculo al que pertenece el valle (hacia la derecha) para intentar mantener el valle (figura F.3h,i), pero no es suficiente y el espacio de *velocidades superior* en las figuras F.3k,l se cierra, pasando a la situación *Frenar*.

Durante un pequeño periodo de tiempo el sistema de navegación considera sólo el objeto *obj1* (figura 5.6o), hasta que se evita y aparece únicamente el *obj2* en la figura F.4b,c) pasándolo por detrás. Durante ese tiempo, el *obj3* ha pasado también y no influye en la navegación. A continuación, resuelve el problema con *obj4* evitándolo por detrás y finalmente, con *obj5* pasando por delante de él a máxima velocidad (figuras 5.6e-l).

F.3. Adelantamiento

En la figura F.5 aparecen imágenes del *ET* y el *EVD* a lo largo del *adelantamiento*. El *EVD* de la figura F.5b muestra únicamente dos zonas rojas en la parte más alta, correspondientes a los dos objetos estáticos que representan los límites de la carretera, por lo que el robot puede circular por su carril libremente. Cuando se acerca al *obj1*, éste aparece mapeado en el *EVD* y el robot inicia la maniobra de adelantamiento evitando al *obj1* por la izquierda. Esto se ve en las figuras F.5c-f. Una vez lo adelanta, el robot debe volver a su carril para no colisionar con el *obj2*, el cuál aparece mapeado en el *EVD* de la figura 5.9h ocupando velocidades de giros a izquierda, sin interferir en la trayectoria del robot.

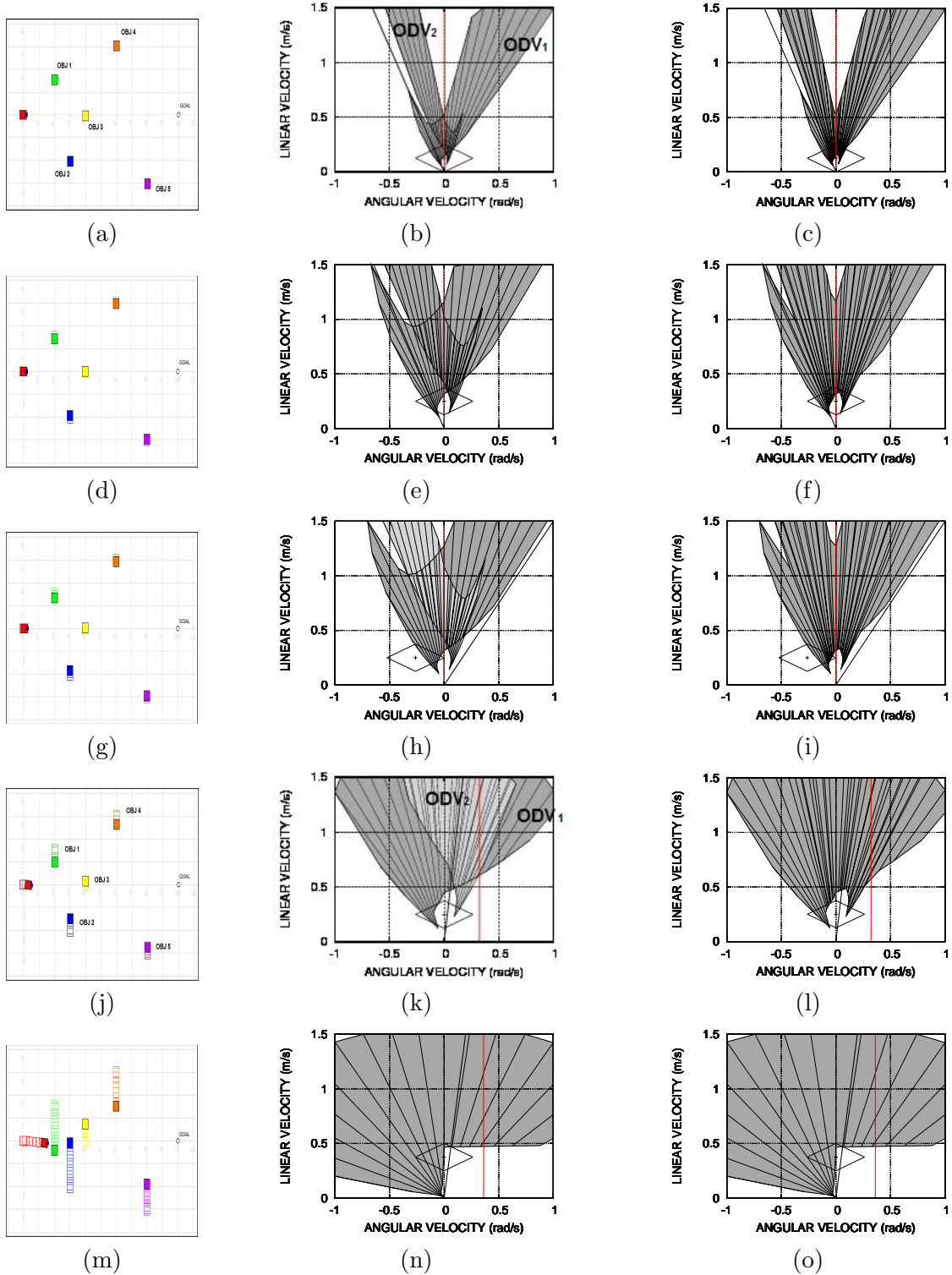


Figura F.3: Evolución del ET y del EVD durante la simulación *carriles*.

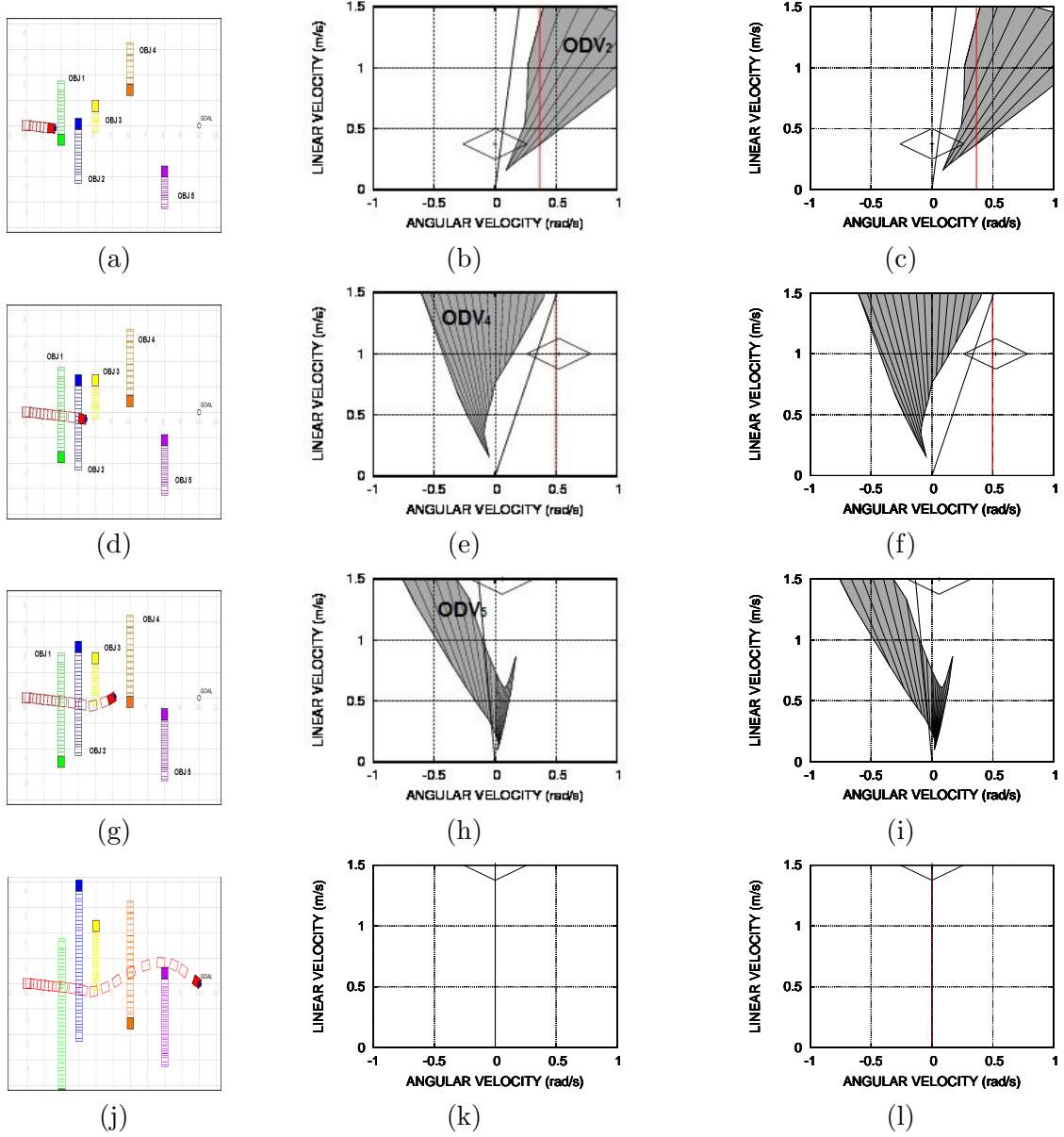


Figura F.4: Continuación de la evolución del ET y del EVD durante la simulación *carriles*.

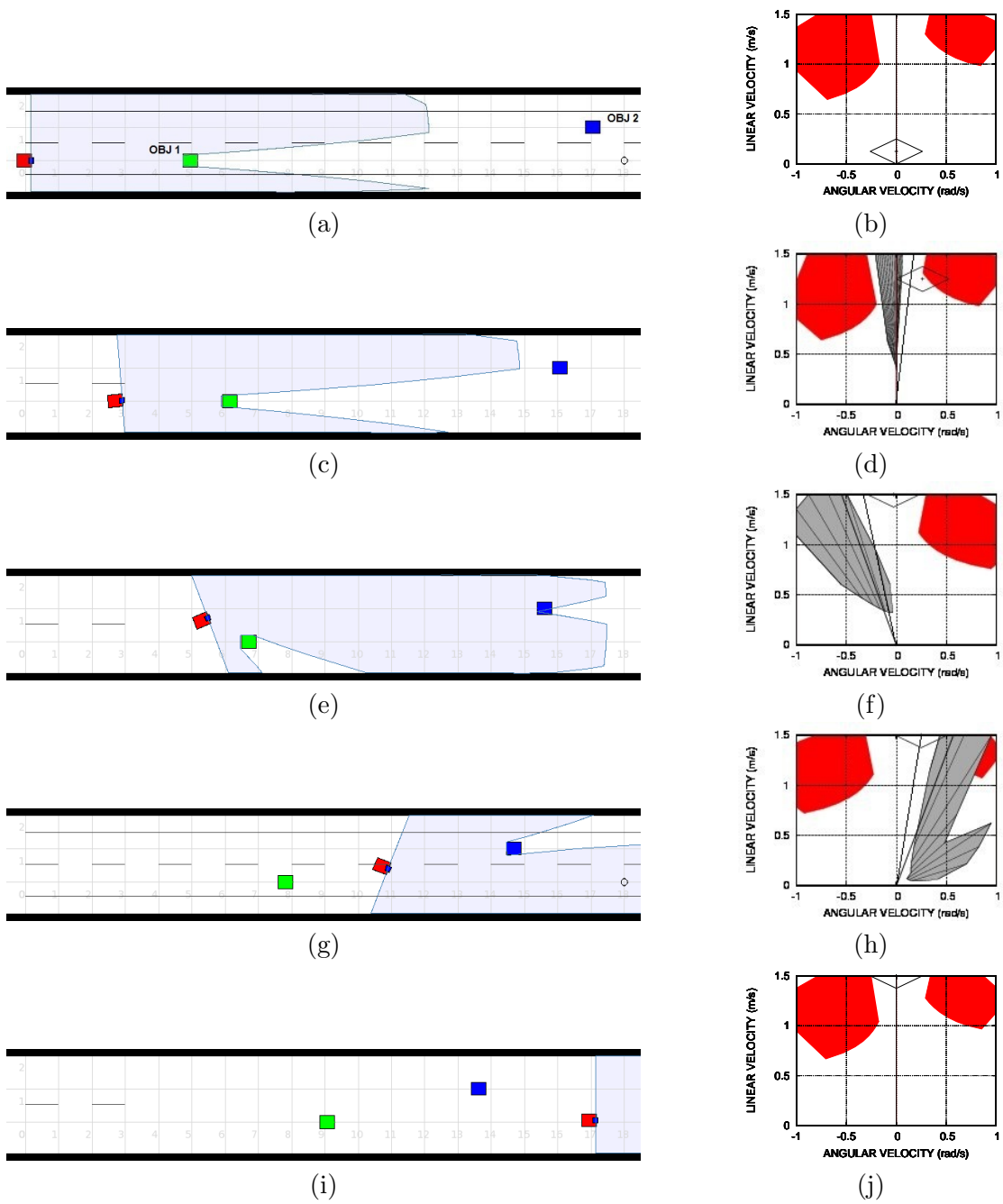


Figura F.5: Evolución del ET y del EVD durante la simulación *adelantamiento*.

Apéndice G

Experimentos

G.1. Experimento 1

Inicialmente, al no detectarse ningún obstáculo dinámico (figura G.1a), el robot puede navegar libremente alcanzando máxima velocidad lineal mientras que la velocidad angular permanece invariable. En el momento en el que se detecta el obstáculo *obj1*, el robot empieza una maniobra de acuerdo a la situación *PasarPrimero* (figura G.1b). Tras varias iteraciones, se detecta un nuevo obstáculo, *obj2*, que lleva al planificador a seguir con *PasarPrimero* como resultado del análisis de la fusión del *EVD*. Sin embargo, en la figura G.1c se puede observar la detección de un falso obstáculo dinámico que hace que se identifique una situación en la que el robot está dentro de su *BC*. Dado que hay velocidades superiores libres, el robot puede seguir un criterio hacia el *goal*, verificando que el comando calculado no caiga en zona ocupada. Cuando desaparece, únicamente están visibles los obstáculos *obj1* y *obj2*, dando como resultado un *ODV* fusionado con un valle que abarca gran parte del *EVD* de la figura G.1d. La situación es *PasarPrimero*, que llevará al robot a pasar por delante de los dos obstáculos. Una vez superado el *obj1*, debe evitar el *obj2*, representado en G.1e, por lo que se escoge *EvitandoObjeto* hasta que se supera y la situación pasa a *MovimientoLibre* (figura G.1f).

Finalmente, se detecta el *obj3* (figura G.2a). Las estrategias varían entre *MovimientoLibre* y *PasarPrimero* por desconvergencia del filtro en los primeros pasos, lo que lleva al robot a tomar velocidades angulares positivas que se traducen en un giro hacia la izquierda a máxima velocidad angular (figuras G.1b,c). Como consecuencia, el objeto dinámico en el *EVD* evoluciona pasando a ocupar velocidades de giros a derecha (lado izquierdo del *EVD*) (figuras G.1d,e), conduciendo al planificador a seleccionar la estrategia *EvitandoObjeto* que permite al robot terminar de resolver la navegación (figura G.1f) y alcanzar el *goal*. La duración total del experimento fue de unos 42 segundos.

G.2. Experimento 2

Inicialmente, tal y como se refleja en la figura G.3a, el robot puede navegar libremente porque no se detecta ningún obstáculo dinámico, llegando a alcanzar la velocidad máxima.

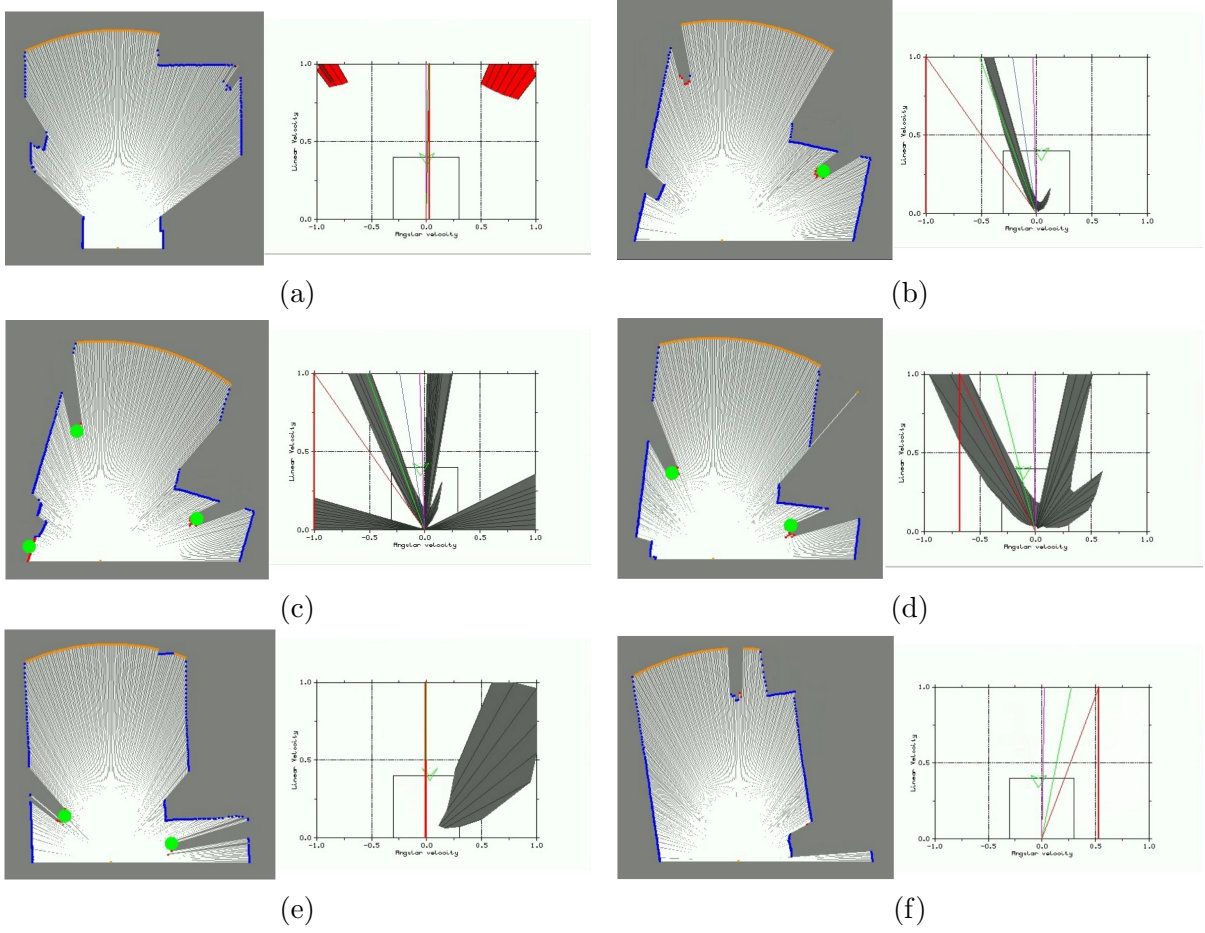


Figura G.1: Experimento 1 en el escenario tipo hall con información sobre el entorno y el *EVD* resultante.

Tras pocas iteraciones, la nube de puntos estáticos en frente del robot aparece mapeada en el *EVD* de la figura G.3b restringiendo el conjunto de *velocidades superiores* y llevándolo a modificar su trayectoria con un giro hacia la izquierda, manteniendo un valor de velocidad lineal máximo. La situación identificada hasta ese momento es la de *Movimiento Libre*. Sin embargo, este objeto estático es en realidad el objeto dinámico *obj1*. Puesto que la velocidad del robot y del obstáculo es pequeña y ambos se mueven en la misma dirección, el software de clasificación de información estática y dinámica no detecta que se trata de información dinámica. No obstante, el sistema de planificación selecciona la estrategia adecuada para resolver la situación.

A continuación, el *obj2* se aproxima por el carril izquierdo (círculo verde de la figura G.3c). La situación elegida por el planificador varía entre *Movimiento Libre* y *Evitando Objeto*, según la representación de *obj2* sobre el *EVD*, ya que todavía queda libre un pequeño conjunto de velocidades en el lado izquierdo del *EVD* de la figura G.3c, que llevan al robot a recuperar su

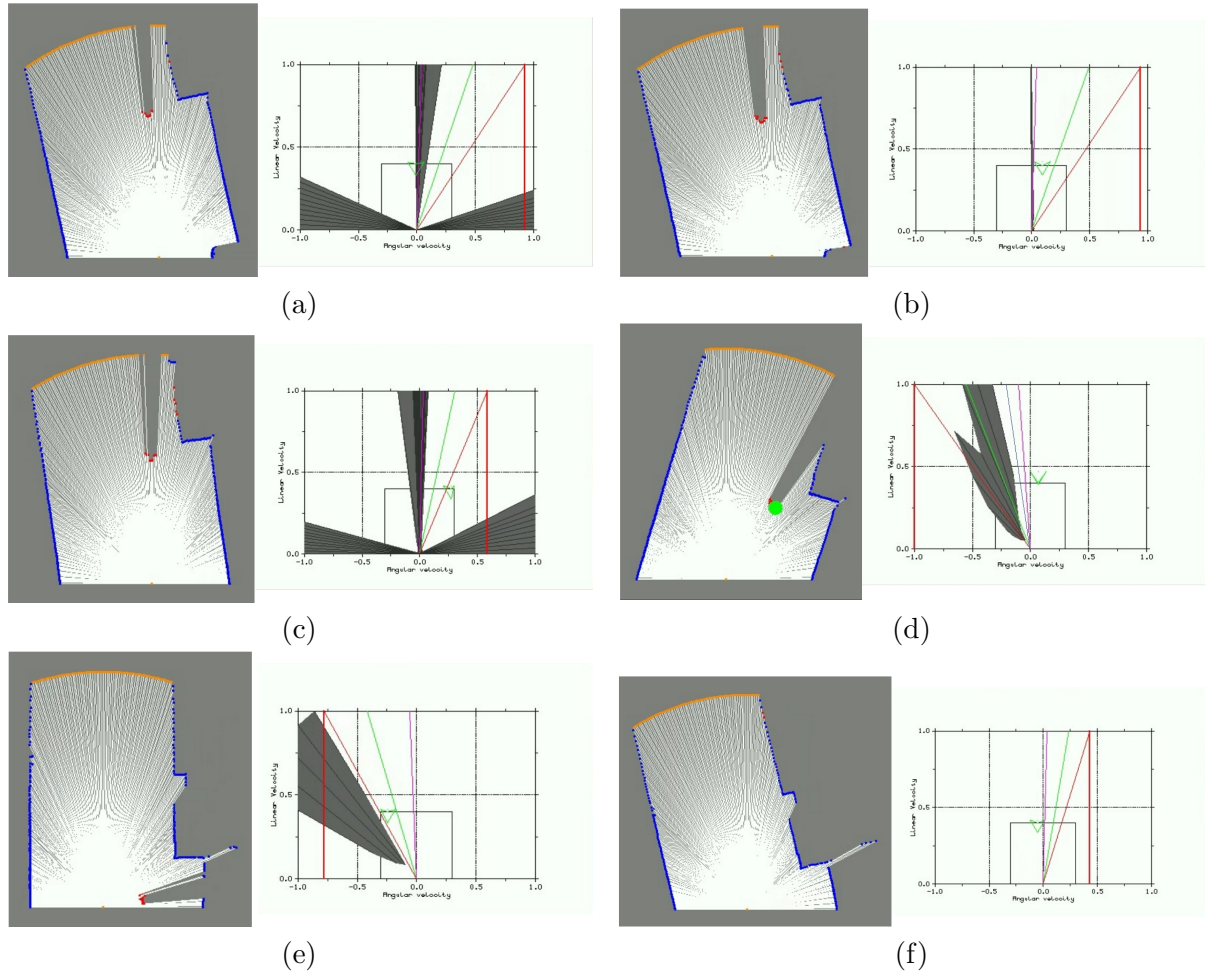


Figura G.2: Continuación del experimento 1 con información sobre el entorno y el EVD resultante.

posición en el carril derecho. En el momento en el que se cierra completamente el espacio de velocidades superiores (figura G.3d), el robot pasa a ejecutar una estrategia *Frenar* reduciendo su velocidad lineal para esperar a que el obstáculo *obj2* pase. Conforme el *obj2* se mueve, el *EVD* se va liberando (figura G.3e) hasta que desaparece completamente de la visibilidad del robot (figura G.3f). La situación pasa a ser *MovimientoLibre* y se ejecutan comandos de velocidad angular positiva. Esto hace que el robot modifique su dirección de movimiento con respecto al *obj1*, el software lo clasifica como dinámico y el *EVD* aparece completamente ocupado (figura G.4a), por lo que el robot debe frenar. Este movimiento hace que se vuelva a detectar como estático en la figura G.4b. Por eso, durante esta etapa del adelantamiento se seleccionan las situaciones *MovimientoLibre*, *Frenar* y *EvitandoObjeto* según si el *obj1* se identifica como estático o dinámico a medida que el robot avanza. Así, hasta que el robot lo adelanta (figura G.4c) y se pasa a una situación *MovimientoLibre*, recuperando su posición en el carril derecho (figuras G.4d,e).

Más adelante, se detecta un nuevo obstáculo *obj3* en el carril izquierdo que no interfiere en la navegación del robot. En la figura G.4f se ve que el *goal* y la dinámica del robot están en zona libre. La situación es *EvitandoObjeto* hasta que pasa a *MovimientoLibre* cuando el *obj3* desaparece, figura G.5a. Entonces, se detecta el *obj4* aproximándose hacia el robot por su mismo carril (figuras G.5b,c). Las situaciones son *MovimientoLibre* y *EvitandoObjeto* para evitar el obstáculo y alcanzar finalmente el *goal*. En esta última etapa del experimento, las situaciones varían entre *MovimientoLibre* y *Frenar* porque el *goal* se encuentra próximo a una de las paredes. Los perfiles de velocidad muestran cambios en sus valores, combinando movimientos angulares de rectificación hacia el *goal* y lineales para acercarse a él (figuras G.5d,e). El tiempo de ejecución del experimento ha sido de unos 66 segundos.

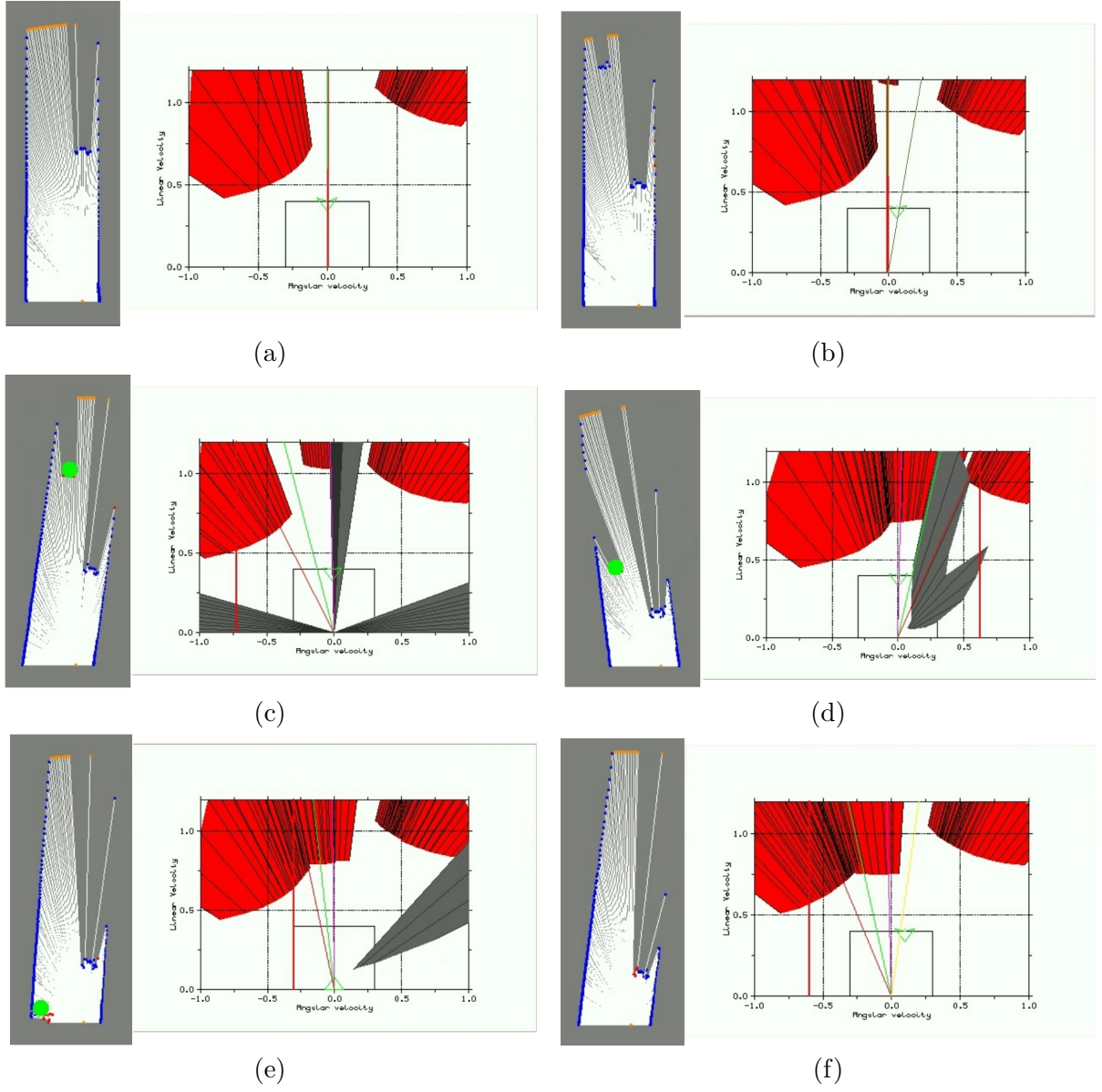


Figura G.3: Experimento 2 en el pasillo con información sobre el entorno y el *EVD* resultante.

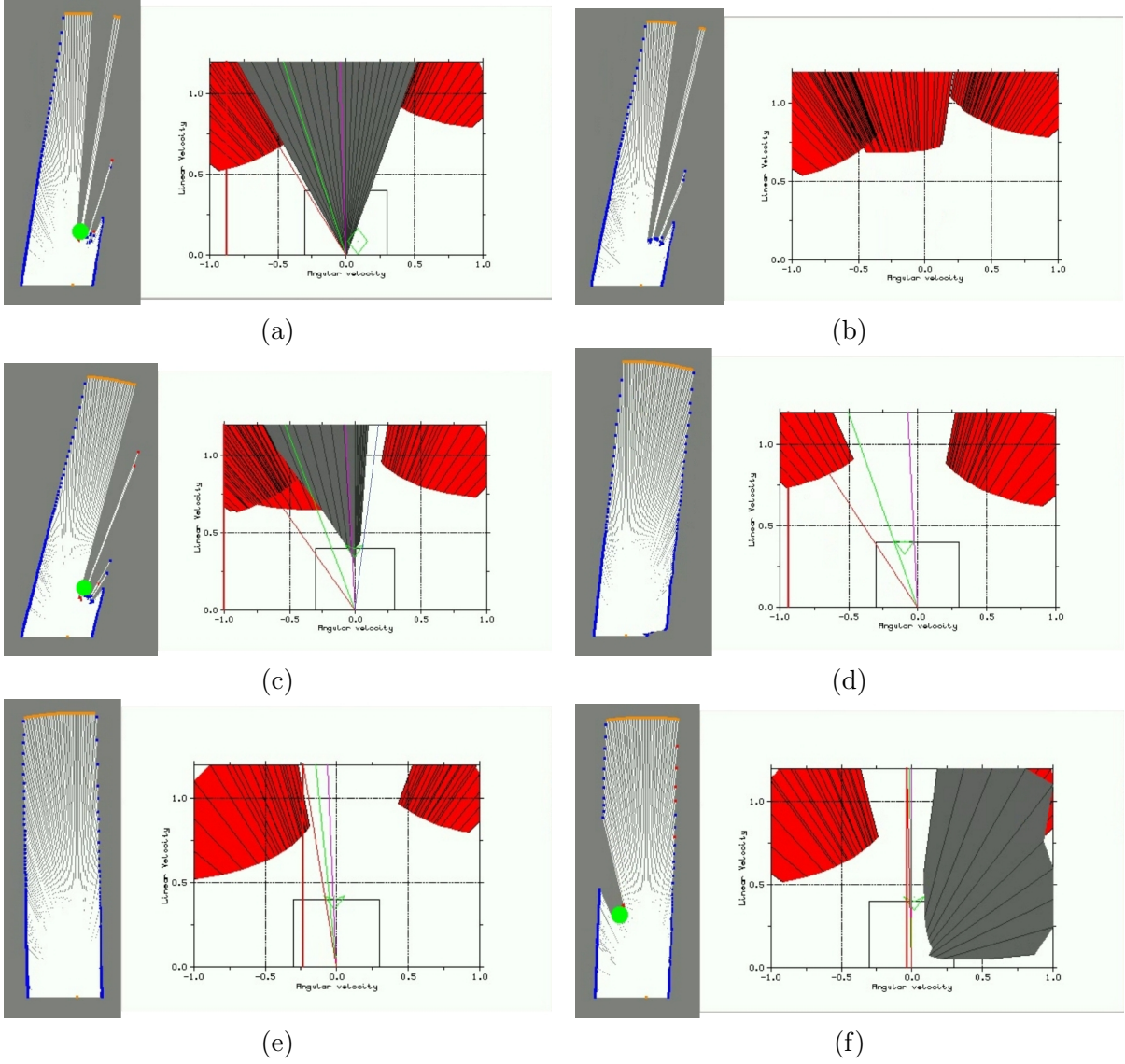


Figura G.4: Continuación del experimento 2, con información sobre el entorno y el *EVD* resultante.

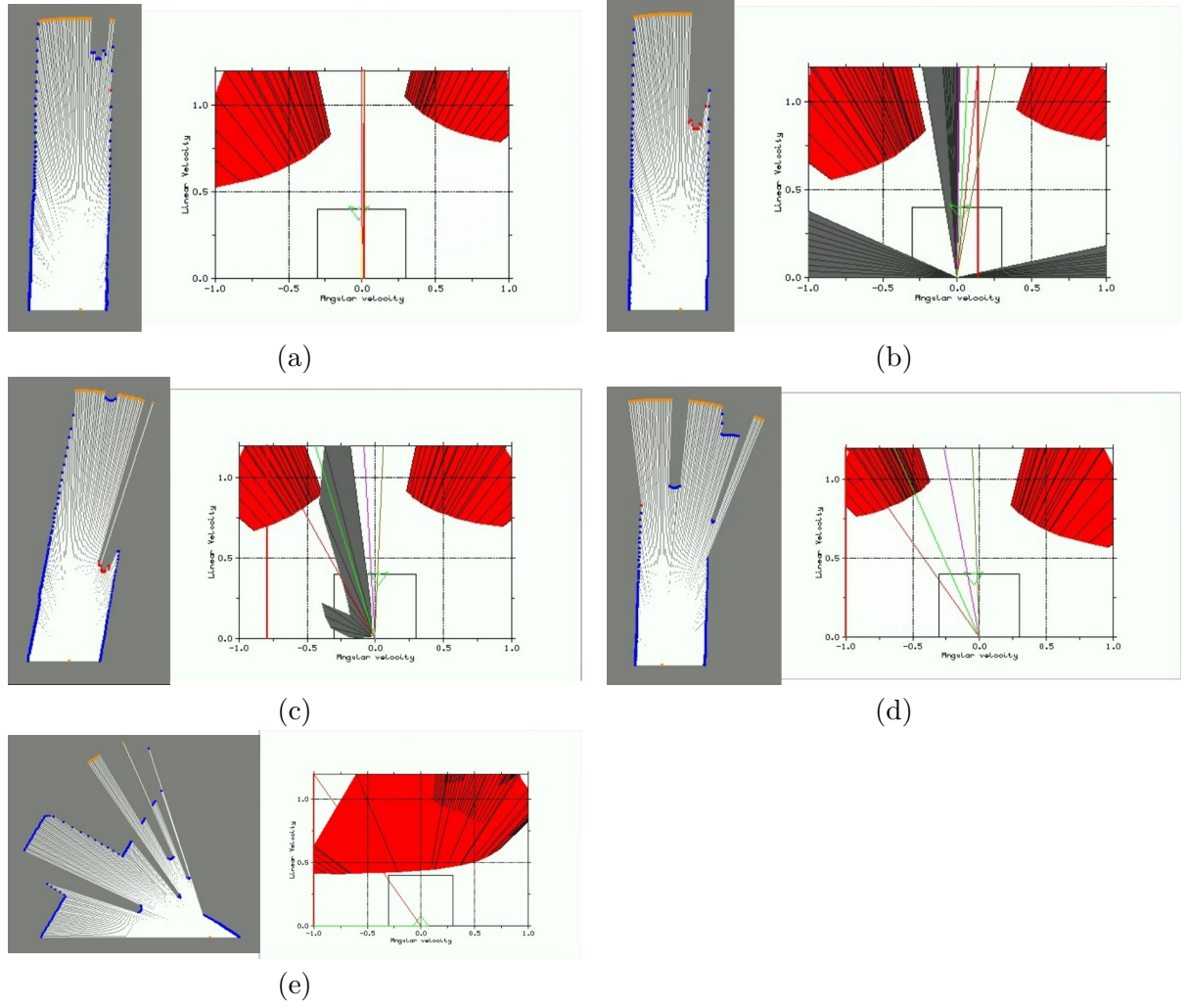


Figura G.5: Continuación del experimento 2, con información sobre el entorno y el *EVD* resultante.

Apéndice H

Distribución temporal

En este apéndice se explica la distribución temporal de cada una de las partes de este PFC. La figura H.1 muestra el Diagrama de Gantt donde se reflejan las diferentes tareas realizadas y el tiempo dedicado a su desarrollo.

La fase de documentación consistió en la lectura de los diferentes trabajos previos relacionados con el tema de este PFC, así como de diferente información sobre temas relacionados con las tareas a realizar.

La siguiente etapa se dedicó a analizar el código de partida sobre el modelo implementado en Matlab para proponer en la siguiente fase un nuevo diseño del software del sistema en C/C++ que supuso el análisis y estudio del modelo en profundidad. Se realizaron varias pruebas para comprobar su correcto funcionamiento.

La siguiente fase consistió en analizar la evolución del modelo en diferentes escenarios y las posibles maniobras de actuación para definir un conjunto de situaciones y estrategias inicial. A continuación, hubo que considerar la fusión de los objetos móviles y el tratamiento de la información estática para elaborar el conjunto final de estrategias.

Para probar el sistema se realizaron varias simulaciones en C/C++ durante la siguiente fase.

En la siguiente etapa se pasó a trabajar con [Player/Stage], dedicando tiempo a aprender a manejar el entorno de trabajo. Paralelamente, se invirtió tiempo en familiarizarse con el software sobre tratamiento de la información del sensor y el *FKE*.

La fase de pruebas con robots reales consistió en una primera etapa de ajuste de parámetros de los valores de velocidad y aceleración del robot real, para solucionar los problemas encontrados con el control de la velocidad de los robots, seguida de los experimentos reales que demuestran el funcionamiento del sistema.

La realización de la memoria se inició una vez terminada la fase de experimentación. Además, aunque no se vea reflejado en el diagrama, al mismo tiempo se escribió un artículo con los resultados obtenidos.

La dedicación a la realización del PFC fue de una media de 6h/día, excepto a partir de la fase de simulación, cuya dedicación fue de unas 2h/día para compaginarlo con el mundo laboral.



Figura H.1: Diagrama de Gantt.

Índice de figuras

| | |
|--|----|
| 2.1. Modelado de objetos móviles en el <i>EVD</i> | 7 |
| 2.2. Fusión de objetos móviles sobre el <i>EVD</i> | 8 |
| 2.3. Modelado de objetos estáticos en el <i>EVD</i> | 9 |
| 3.1. Toma de decisión | 12 |
| 3.2. Árbol de situaciones | 13 |
| 4.1. Situación <i>MovimientoLibre</i> | 17 |
| 4.2. Situación <i>PasarPrimero</i> | 18 |
| 4.3. Situación <i>PasarAlineado</i> | 19 |
| 4.4. Situación <i>Frenar</i> | 19 |
| 4.5. Situación <i>EvitandoObjeto</i> | 20 |
| 4.6. Dentro de la banda de colisión | 21 |
| 4.7. Situación <i>ColisionSegura</i> | 21 |
| 4.8. Situación con varios obstáculos. <i>PasarPrimero</i> | 22 |
| 4.9. Situación con varios obstáculos. <i>Frenar y PasarPrimero</i> | 22 |
| 5.1. Simulación <i>caos</i> . Perfiles de velocidad | 24 |
| 5.2. Simulación <i>caos</i> . Perfil de situaciones | 25 |
| 5.3. Simulación <i>caos</i> . <i>ET</i> | 25 |
| 5.4. Simulación <i>carriles</i> . Perfiles de velocidad | 26 |
| 5.5. Simulación <i>carriles</i> . Perfil de situaciones | 26 |
| 5.6. Simulación <i>carriles</i> . <i>ET</i> | 27 |
| 5.7. Simulación <i>adelantamiento</i> . Perfiles de velocidad | 28 |
| 5.8. Simulación <i>adelantamiento</i> . Perfil de situaciones | 28 |
| 5.9. Simulación <i>adelantamiento</i> . <i>ET</i> y <i>EVD</i> | 29 |
| 6.1. Experimento 1. Perfiles de velocidad | 32 |
| 6.2. Experimento 1. Perfil de situaciones y trayectoria del robot | 33 |
| 6.3. Experimento 1. Capturas de la prueba en real | 33 |
| 6.4. Experimento 2. Perfiles de velocidad | 35 |
| 6.5. Experimento 2. Perfil de situaciones y trayectoria del robot | 35 |
| 6.6. Experimento 2. Capturas de la prueba en real | 36 |
| E.1. Variables de decisión | 55 |

| | |
|---|----|
| E.2. Situación <i>MovimientoLibre</i> . <i>ET</i> y <i>EVD</i> | 56 |
| E.3. Situación <i>PasarPrimero</i> | 58 |
| E.4. Situación <i>PasarAlineado</i> | 59 |
| E.5. Situación <i>Frenar</i> | 61 |
| E.6. Situación <i>EvitandoObjeto</i> | 62 |
| E.7. Dentro de la banda de colisión | 63 |
| E.8. Dentro de la banda de colisión (2) | 63 |
| E.9. Situación <i>ColisionSegura</i> | 64 |
| E.10. Varios obstáculos. Situación <i>PasarPrimero</i> | 65 |
| E.11. Varios obstáculos. <i>PasarPrimero</i> y <i>Frenar</i> (1) | 66 |
| E.12. Varios obstáculos. <i>PasarPrimero</i> y <i>Frenar</i> (2) | 67 |
| | |
| F.1. Simulación <i>caos</i> . <i>ET</i> y <i>EVD</i> (1) | 70 |
| F.2. Simulación <i>caos</i> . <i>ET</i> y <i>EVD</i> (2) | 71 |
| F.3. Simulación <i>carriles</i> . <i>ET</i> y <i>EVD</i> (1) | 73 |
| F.4. Simulación <i>carriles</i> . <i>ET</i> y <i>EVD</i> (2) | 74 |
| F.5. Simulación <i>adelantamiento</i> . <i>ET</i> y <i>EVD</i> | 75 |
| | |
| G.1. Experimento 1. Información capturada por el láser y <i>EVD</i> (1) | 78 |
| G.2. Experimento 1. Información capturada por el láser y <i>EVD</i> (2) | 79 |
| G.3. Experimento 2. Información capturada por el láser y <i>EVD</i> (1) | 81 |
| G.4. Experimento 2. Información capturada por el láser y <i>EVD</i> (2) | 82 |
| G.5. Experimento 2. Información capturada por el láser y <i>EVD</i> (3) | 83 |
| | |
| H.1. Diagrama de Gantt | 87 |

Bibliografía

- [Asensio y Montano] J.R. Asensio and Luis Montano. A kinematic and dynamic model-based motion controller for mobile robots. *IFAC World Congress*, 2002.
- [Balkcom y Mason] D. Balkcom and M. Mason. Time optimal trajectories for bounded velocity differential drive vehicles. *International Journal of Robotic Research*, 21(3):199–218, 2002.
- [Robotica] Aníbal Ollero Baturone. *Robótica: manipuladores y robots móviles*. Marcombo Boixareu, D.L. 2001.
- [Brock y Kathib] O. Brock and O. Kathib. High speed navigation using the global dynamic window approach. In *Proc. Int. Conf. on Robotics and Automation*, pages 1238–43, 1999.
- [Castro y otros] D. Castro, U. Nunes, and A. Ruano. Reactive local navigation. In *IEEE Int. Conference on Electronic and Industrial IECON*, pages 2427–2432, Sevilla, 2002.
- [Choset y otros] H. Choset, K.M Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, , and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [Fiorini y Shiller] P. Fiorini and Z. Shiller. Robot motion planning in dynamic environments. In *International Symposium of Robotic Research*, pages 237–248, 1995.
- [Fox y otros] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.
- [Fraichard] Th. Fraichard. Trajectory planning in a dynamic workspace: a state-time-space approach. *Advanced Robotics*, 13(1), 1999.
- [Player/Stage] <http://playerstage.sourceforge.net>. Proyecto *Player/Stage*.
- [CGAL] <http://www.cgal.org>. Librería de algoritmos geométricos *CGAL*.
- [DISLIN] <http://www.dislin.de>. Librería gráfica *DISLIN*.
- [Ho y Simmons] N.Y. Ko and R. Simmons. The lane-curvature method for local obstacle avoidance. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, 1998.

- [Kuffner y LaValle] J. Kuffner and S. LaValle. Randomize kinodynamic planning. *International Journal of Robotics Research*, (6):378–400, 2001.
- [Latombe] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Press, 1990.
- [Minguez y Montano] J. Minguez and L. Montano. Sensor-based motion generation in unknown, dynamic and troublesome scenarios. *Robotics and Autonomous Systems*, 52(4):290–311, 2005.
- [Montesano y otros] L. Montesano, J. Minguez, and L. Montano. Modeling dynamic scenarios for local sensor-based motion planning. *Autonomous Robots*, 25(3):231–251, 2008.
- [Owen y Montano (2005)] E. Owen and L. Montano. Motion planning in dynamic environments using the velocity space. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 997–1002, 2005.
- [Owen y Montano (2006)] E. Owen and L. Montano. A robocentric motion planner for dynamic environments using the velocity space. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4368–4374, 2006.
- [Owen, Lorente y Montano] Eduardo Owen, María T. Lorente, and Luis Montano. Velocity space situations: a robocentric planner for safe navigation in dynamic environments. *Autonomous Robots*. En revisión.
- [Renaud y Fourquet] M. Renaud and J-Y. Fourquet. Minimum time motion of a mobile robot with two independent, acceleration-driven wheels. In *IEEE Int. Conference on Robotics and Automation*, pages 2608–2612, Albuquerque, New Mexico, 1997.
- [Santos y otros] Sergio Santos, José Eduardo Faria, Fernando Soares, Rui Araújo, and Urbano Nunes. Tracking of multi-obstacles with laser range data for autonomous vehicles. *Proceeding of ROBOTICA 2003 - Scientific ROBMeeting of the 3rd Robotics National Festival*, 2003.
- [Stachniss y Burgard] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 997–1002, 2002.
- [Welch y Bishop] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina, Department of Computer Science.